April 1981
USA $2.95/DM 9.80

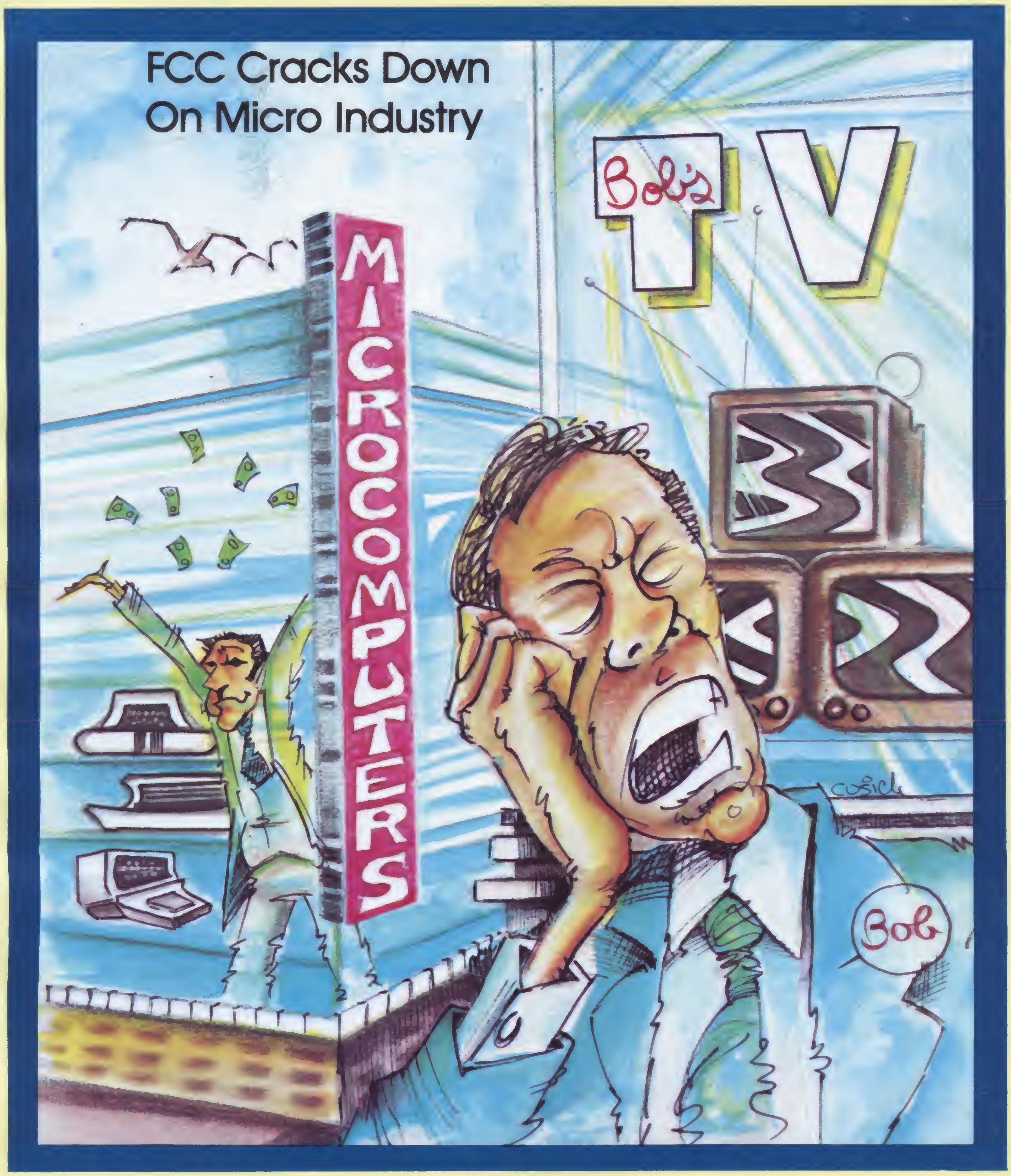
kilobaud

MICROCOMPUTING

T.M.

DESK-TOP COMPUTERS

Cleaning the Air of RFI



Micros Say the Darn'dest Things □ Pascal Means Business □ Energize Those OSI Peripheral Ports □ Quicksilver Micro Kit Monitors the Mercury □ Word Wizardry with WordStar □ Crack That Code with the Apple II □ The Atari Shows Promise □ The Ultimate 6800 Information Juggler

0 74470 65945 04

MICROCOMPUTING

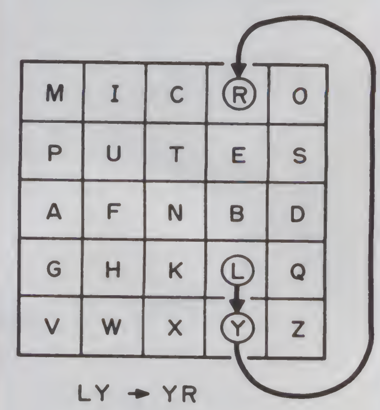
PUBLISHER/EDITOR
Wayne Green
EXECUTIVE VICE PRESIDENT
Sherry Smythe
EDITORIAL MANAGER
Jeff DeTray
PUBLICATIONS MANAGER
Edward Ferman
MANAGING EDITOR
Dennis Brisson
ASST. MANAGING EDITOR
Susan Gross
COPY EDITOR
Eric Maloney
EDITORIAL ASSISTANT
Linda Stephenson
ADMINISTRATIVE ASSISTANTS
Pat Graham, Nancy Noyd
ASSOCIATE EDITORS
Robert Baker, Ken Barbier, Frank Derfler, Jr., Rod Hallen, Peter Stark, Sherm Wantz

MANUFACTURING MANAGER
Noel Self
PRODUCTION MANAGER/PUBLICATIONS
Nancy Salmon
ASST. PRODUCTION MANAGER
Michael Murphy
ART DIRECTOR
Diana Shonk
ADVERTISING GRAPHICS
Robert Drew, Bruce Hedin, Steve Baldwin
PRODUCTION DEPARTMENT
Joan Ahern, William Anderson, Jr., Linda Drew, Bob Dukette, Kenneth Jackson, Ross Kenyon, Maryann Metivier, Theresa Ostebo, Jane Preston, Patrice Scribner, Suzanne Self, Susan Symonds, Thomas Villeneuve
PHOTOGRAPHY
William Heydolph, Terrie Anderson, Bill Suttenfield
TYPESETTING
Barbara Latti, Sara Bedell, Michele DesRochers, Luann Keddy, Mary Kinzel, Linda Locke, Karen Stewart

CORPORATE CONTROLLER
Charles Garniss, Jr.
EXECUTIVE ASSISTANT
Leatrice O'Neil
ACCOUNTING MANAGER
Knud Keller
CIRCULATION MANAGER
Debra Boudrieau
CIRCULATION
Doris Day, Pauline Johnstone, Dion Owens, designer
BULK SALES MANAGER
Ginnie Boudrieau
ASSISTANT TO THE PRESIDENT
Matthew Smith
ADVERTISING
603-924-7138
Kevin Rushalko, Mgr.
Louise Caron, Dick Hayes, Hal Stephens, Marcia Stone

Page 30.

Page 64.



Contents: April 1981

Volume V
No. 4

This month:

You can always tell when a fledgling industry has finally made the big time: the Federal Government finds a way to regulate it. Sure enough, it's happened to the microcomputer industry in the form of regulations governing radio frequency interference (RFI).

In this month's cover story ("FCC Takes Aim Against RFI Polluters"), we examine the Federal Communications Commission's rules on computer-generated RFI, delve into the history of the RFI controversy and explore the ramifications of the new regulations on both manufacturers and users. There are many questions to be considered.

Why, for instance, are the RFI regulations for microcomputers many times more stringent than for minicomputers and mainframes? Why are some major firms meeting the RFI regulations with ease, while others struggle to meet already extended deadlines? And what about the smaller firms making peripherals and accessories? Can they cope with the FCC's costly RFI certification mandate?

It may be tempting to believe that we, as microcomputer users, need not concern ourselves with the problems of manufacturers. Unfortunately, things are more complicated than that. We may find that as a result of the RFI rules, our micros will become more costly and less flexible than in the past. But judge for yourself; the story begins on page 30.

—The Editors

Next month:

Kilobaud Microcomputing will explore the world of word processing in next month's issue. You'll examine the many software systems available to turn your computer into an efficient word processing machine and discover new applications as some of our readers share their word processing experiences.

This month's cover:

Illustration by Richard Cusick.

Page 84.

Page 174

Kilobaud *Microcomputing* (ISSN 0192-4575) is published monthly by Wayne Green, Inc., 80 Pine St., Peterborough NH 03458. Subscription rates in U.S. are $25 for one year and $53 for three years. In Canada: $27 for one year only, U.S. funds. Foreign subscriptions (surface mail)—$35 for one year only, U.S. funds. Foreign air mail subscriptions —$62 for one year only, U.S. funds. Canadian Distributor: Micron Distributing, 409 Queen St. West, Toronto, Ontario, Canada M5V 2A5. In Europe, contact: Monika Nedela, Markstr. 3, D-7778 Markdorf, W. Germany. South African Distributor: KB Microcomputing, PO Box 782815, Sandton, South Africa 2146. Australian Distributor: Electronic Concepts, Attention: Rudi Hoess, 55 Clarence Street, Sidney 2000, Australia. Second-class postage paid at Peterborough NH 03458 and at additional mailing offices. Phone: 603-924-3873.

PUBLISHER'S REMARKS

By Wayne Green

Short-lived Computer Games

Ballyhoo

Having just returned from another visit to the biannual madhouse known as CES (Consumer Electronics Show) in Las Vegas, I thought you might like to know what you missed: not much.

I thought the single outstanding event was the showing of the new television set by Panasonic. The darned thing is about the size of a pocket calculator, complete with an LCD screen about two by three inches. The picture wasn't very good, but it *was* there. It could probably be used as a screen for a pocket computer.

Other than that, microcomputer enthusiasts were hard put to experience much joy. Commodore showed their new color system, the VIC-20, which was indeed impressive and may have Radio Shack going back to the lab to increase the performance of their color system. But I got the impression that Commodore was thinking largely in terms of games, though giving some lip service to business applications.

Atari was there—complete with a very nice buffet—to express their hopes for the future. Again it was games and more games, with a mention of business. The Bally exhibit was almost totally game-ridden, as was the Mattel exhibit—identical to their showing in Chicago at the summer CES. I can understand the thinking which is involved here. Both Atari and Bally have come from the arcade-games business and are used to people plunking down substantial sums of money to play computerized arcade games. But I am wondering just how long this enthusiasm will continue.

While I admit that I can't hold myself up as a typical games person, I enjoy playing games in an arcade as much as the next guy; I flick in the quarters and jump around yelling as loud as anyone. But then when I get the same game set up at home, the thrill goes out of it in a matter of a couple of days. Perhaps having the game at home is like sex in marriage, which seldom has the excitement of the pre-marital chase.

The fact is that I am far more, not less, into games than most people. I will drop everything when *New York* magazine arrives so I can tackle their damned crossword puzzle. I love cryptograms and use my Casio wrist watch to time my solutions. I enjoy the Master-Mind-type of puzzles and time my solutions to those, too. All my life I've enjoyed card games, and I'm very good at them. I have yet to find anyone who can beat me at Cribbage.

One of the things that makes me difficult to beat—besides a developed sense of conniving fiendishness needed to win by wits—is my consistent luck. This luck holds true not only in games, but in life too. Oh, I've had some heavy traumas, but each of them came from putting my trust in people, not in having bad luck. I would be hard put to attribute any serious problems to luck, or lack of it.

I developed this interest in games early in life because my whole family was game oriented, and most evenings were spent in playing card and board games. Still, with all this gamey background, I find that computer games bore the hell out of me in a few hours, no matter how much fun they are at first.

There is no shortage of remarks in the papers about the short life of the early television games when brought into the home, and we've found the same thing with our computerized games. We seem to get much of our enjoyment of games from their novelty, from being out in public and perhaps even from paying for them.

When I first became interested in computers, I was dying to have a real chance to have at Hammurabi and Star Trek. Hammurabi lasted about two days, and Star Trek perhaps four, before I became bored. I've never played either of them since. They are as important to me as the paddles, which are gathering dust under the TV set, and the mock rifle for shooting TV targets. I had fun with those for maybe a week before putting them up permanently.

Is there any reason why even the great games now available on the Bally, Mattel and Atari computers are going to last more than a few days before boredom sets in? And what comes next? Is it possible that this boredom with these incredibly expensive toys won't be considered news by newspapers and magazines, in turn scuttling sales for the game purveyors? I'd be surprised if *The Wall Street Journal* hasn't already lined up its sights on this fat target. And you know that *Money* and a few other magazines will be hitting hard. *Fortune*, where *are* you?

Has anyone seen a study showing that these expensive computer games last any longer with users than the old home pong games? Playing stickman baseball or soccer is *not* the same as watching a game on TV, and I predict that unless the game people start reviewing their situation, we may be heading for another round of disasters. We've watched the mighty fall by misreading the market, and have seen Mits, Imsai, Processor Tech, The Digital Group, etc., go down the tubes.

Is there any way for all of these game-oriented firms to start hedging their bets? I realize that it is difficult for the entire top management of firms that have made millions on arcade games to even momentarily consider the possibility that home computer games will not be as lastingly popular as arcades, but this may be a thought which has to be thought.

What can they do? One of the best bets is to sit down and talk with some industry consultants—people who are not brainwashed by arcade successes, but who have some perspective on the microcomputer industry, where it started, where it traveled getting to where it is now and where it may be going. Then the trick is to evaluate the hardware available and see what, if anything, can be done to start hedging bets, if this seems prudent.

Yes, there is going to be a big business market, but can the Bally computer adapt to it? And, yes, there is going to be a huge educational market, which may also be difficult to cope with. And eventually there is going to be some kind of a

META TECHNOLOGIES

26111 Brush Avenue, Euclid Ohio 44132
CALL TOLL FREE 1-800-321-3552 TO ORDER
IN OHIO, call (216) 289-7500 (COLLECT)

FILE BOX

DISKETTE STORAGE SYSTEM

$24.95 for 5¼" disks

$29.95 . . . for 8" disks.

MTC brings you the ULTIMATE diskette storage system, at an affordable price. Storing 50 to 60 diskettes, this durable, smoke-colored acrylic unit provides easy access through the use of index dividers and adjustable tabs. Unique lid design provides dust-free protection and doubles as a carrying handle.

'RINGS' & THINGS

Help prevent data loss and media damage due to improper diskette centering and rotation with the FLOPPY SAVER™ reinforcing hub ring kit. 7-mil mylar rings install in seconds. Kit is complete with centering tool, pressure ring, 25 adhesive backed hub rings and instructions.

HUB RING KIT for 5¼" diskettes . $10.95
REFILLS (50 Hub Rings) . . . . $ 5.95

Protect your expensive disk drives and your valuable diskettes with our diskette drive head cleaning kit. The kit, consisting of a pair of special "diskettes", cleaning solution and instructions, can be used for 52 cleanings. Removes contamination from recording surfaces in seconds without harming drives.

CLEANING KIT for 5¼" drives . . . $24.95

PLASTIC LIBRARY CASES

(not shown)

An economical form of storage for 10 to 15 diskettes, and is suitable for your bookshelf! Case opens into a vertical holder for easy access.

5¼-inch diskette case $3.50
8-inch diskette case $3.95

Single Sided, Soft-Sectored 5¼-inch; (for TRS-80™) Mini-floppy

DISKETTES

$21.95 box of 10

These are factory fresh, absolutely first quality (no seconds!) mini-floppies. They are complete with envelopes, labels and write-protect tabs in a shrink-wrapped box.

PLAIN JANE™

DISKETTES
The Beautiful Floppy
with the Magnetic Personality™

Thousands of people have switched to this low-cost alternative. These quality diskettes are packaged in a plain white box . . . no fancy printing, fancy names or fancy labels, not even our own (labels cost money). Trust us.

PLAIN JANE™ Diskettes $21.95
10 boxes of 10 . . . . (each box)$21.50

VERBATIM'S PREMIUM DISKETTES AT AFFORDABLE PRICES

DATALIFE™

Seven data-shielding improvements mean greater durability and longer data life. These individually, 100% error-free certified diskettes feature thicker oxide coating, longer-lasting lubricant, improved liner, superior polishing and more! Meets or exceeds IBM, Shugart, ANSI, ECMA and ISO standards. Reinforcing HUB RINGS help prevent data loss and media damage, reducing errors.

Buy the best . . . buy DATALIFE™.

VERBATIM DATALIFE™ DISKETTES
5¼-inch (box of 10)
MD525-01 . . . . $26.95
10 boxes of 10 . . . . (each box)$25.95
8-inch FLOPPIES
Single-Density, FD34-1000 . . $33.95
Double-Density, FD34-8000 . $43.95

MICROPARAPHERNALIA

DISKETTES (box of ten)

5¼" PLAIN JANE™ $21.95
5¼" DATALIFE™ MD 525-01 . . $26.95
8" DATALIFE™ FD34-1000 . . . . $33.95
8" DATALIFE™ FD34-8000 . . . . $43.95

NEWDOS by APPARAT

NEWDOS/80 ★ SPECIAL $129.95
NEWDOS + to
NEWDOS/80 UPGRADE . . . . . CALL
NEWDOS + with ALL UTILITIES
35-track $69.95
40-track $79.95

BOOKS

TRS-80™ DISK AND OTHER MYSTERIES . . $19.95
MICROSOFT™ BASIC DECODED $29.95
1001 THINGS TO DO WITH YOUR PERSONAL COMPUTER . . . . $ 7.95

Let Your TRS-80™ Test Itself With

THE FLOPPY DOCTOR & MEMORY DIAGNOSTIC

by THE MICRO CLINIC

A complete checkup for your Model I. THE FLOPPY DOCTOR completely checks every sector of 35- or 40-track disk drives. Tests motor speed, head positioning, controller functions, status bits and provides complete error logging. THE MEMORY DIAGNOSTIC checks for proper write/read, refresh, executability and exclusivity of all address locations. Includes both diagnostics and complete instruction manual.

SYSTEM DIAGNOSTICS $19.95

An improved version of the SYSTEM DIAGNOSTICS above. Designed for single or double density, 35-, 40-, 77-, or 80-track disk drives. Includes new and modified tests. Features THE FLOPPY DOCTOR, Version 3.0.

SYSTEM DIAGNOSTICS-V3 . . $24.95

CALL FOR INFORMATION ON OTHER PRODUCTS ✓161

TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation. DATALIFE is a trademark of VERBATIM. PLAIN JANE, AIDS-I, AIDS-III, CALCS-III, CALCS-IV, MERGE-III are trademarks of MTC.

MOST ORDERS SHIPPED WITHIN ONE BUSINESS DAY

Products damaged in transit will be exchanged.

PRICES IN EFFECT April 1, 1981 THRU April 30, 1981, Prices, Specificiations, and Offerings subject to change without notice.

WE ACCEPT
- VISA
- MASTER CHARGE
- CHECKS
- MONEY ORDERS
- C.O.D.

8104

- Add $3.00 for shipping & handling
- $3.00 EXTRA for C.O.D.
- Ohio residents add 6½% sales tax.

home market, but how soon, and what will it really be like? And then, are the Mattel and APF aimed right for it? I think I have some good answers to the above questions, as do a lot of other chaps I know. But I'm not going to give them away for $2.95, and neither is anyone else.

The Apple people, not having the arcade background, perceived the dangers of depending on games for an extended success and have been working towards getting into the business market with their system. The Apple II was almost entirely a game machine, so the attempts at billing it as a business system didn't ring very true, despite the nice ads to the contrary. The color was part of the problem. Business computers are black and white (or green), not in full color. I don't think anyone at Apple thought of this, so when business applications were promoted they, too, were in color, and this made it difficult for businessmen to take them seriously.

There certainly *are* uses for color in business programs, but I suggest that the shrewd programmer remember the black-and-white syndrome and use that except where color is really called for. Color can be used to emphasize something, such as an account that is past due. It can indicate the status of an inventory item for quick spotting by the operator. It can also indicate a changed item or new part number.

I predict that there is enough need for color so that all business systems will eventually have to have it. From that viewpoint the change in chip by Radio Shack for their color system is indeed unfortunate for them. This means that the wealth of programs written for the TRS-80 are going to be difficult to convert for use on their color system. The faster-acting machine-language programs are going to have to be rewritten, virtually from scratch. I don't think Radio Shack saw anything but the need for a game system to compete with Bally, Mattel, Atari, Commodore, etc., so they may have not given much consideration to the longer-term needs of the new system.

I predict that there is enough need for color so that all business systems will eventually have to have it.

Blitz Schools

Althought it has been almost 40 years since I entered college, the trauma is still very much alive in my memory. In those days—just after the depression of the 30s—it was a completely accepted dogma that anyone who could afford to went to college. In retrospect, I managed to almost totally waste four years of my life.

My high school tried very hard to be helpful. They brought in professionals to administer aptitude tests to make sure that their graduates were heading toward a career which was best suited to them.

The high school I attended was Erasmus Hall in Brooklyn. It was a large, but fine school at the time. I was dismayed to see a TV program about it the other evening and find that it is now considered the most dangerous school in New York. What a change! We had about 10,000 students and an after-school bunch of activities which made it superb. The school had over 100 different clubs for us to join, each meeting after school.

Each student was given a voice test when entering the school, and the top 100 voices were permitted to join the choral club. Meetings (rehearsals) were during the second period daily, so the group was an extraordinary choral group. The club gave concerts all around New York and was justly famous. Having started singing with a nearby church as a boy soprano, my voice had developed into a good baritone and I got to join this select group.

In addition to that I also was a member of the Savoyards, a group which met one day a week to practice Gilbert & Sullivan operettas. We put on The Mikado before the assembly—a frightening experience with over 7,000 in the audience. As an amateur radio enthusiast, I also joined the school's radio club and helped to operate and work on their ham station, W2ANU. It was one of the longest-lived ham clubs in the country at the time. Then there was the camera club, the book club, the music appreciation club, etc. I was a real joiner. As the current president of the Peterborough Chamber of Commerce, I guess I still am.

What a shame that Erasmus had to go from one of the most beautiful and wonderful of schools to an armed camp, with police needed in all of the halls to prevent mayhem, extortion and drugs. Pity.

My high school aptitude tests showed that I had an enormous mechanical talent. This, coupled with my ham license, indicated that I should be going to a technical college. Having been born in New Hampshire, I had intended on going to Dartmouth and perhaps getting a law degree. But, since my grandfather had become a millionaire in the 20s as an inventor (but subsequently lost everything in the stock market crash in 1929), the results of the tests did not surprise me. I expected heredity to count for something.

I looked over the technical schools and picked the one that seemed to offer the most. It dated back to 1824 and had an excellent reputation. It was awful.

I found myself taking a series of courses which required sheer memorization and no practical application of concepts. This method of study results in a quick loss of concepts learned, without some practical use to reinforce the concepts. Everyone else at the school was in the same position—able to remember little of any course already passed. In many courses, I would have to read the book at night and face a daily quiz on what I'd read. By the time I'd passed the course, all I had left to show for the effort was the grade.

When I was halfway through college, the Japanese made the mistake of screwing around with Pearl Harbor. I found that it was not going to be a simple matter to stay in school. The draft was getting hotter, and I really didn't want to learn about trench warfare. I made contact with a naval commander who was running a research lab in Anacostia, across the river from Washington. He was interested in my ham background and wanted me to join the Navy so I could get into his lab. That sounded OK to me, and certainly better than the pot luck the Army had to offer, so I joined the Navy the day before I was to be inducted into the Army. That was a close call, and one of the better moves I've made in my life.

Before working in the lab, I had to get some background in radar, so I was given a third-class radio technician rating and sent to the Bliss Electrical School in Tacoma Park, on the outskirts of Washington. This was a three-month intro to electricity and radio. I graduated second in my class as a result of my ham background.

The next step was a six-month course in radio and radar at the Radio Materiel School at Treasure Island in San Francisco. This was a superb school, and I learned more about electronics in the six months than my college managed to teach in four years. They had practical courses in theory, followed by lab experience to make sure the theory sunk in and was practical. Every instructor was a ham, and I quickly went from just having a hazy concept of electronics to being able to work with anything electronic with full confidence. Just hand me a schematic and let me at it.

At the end of the school I had to make a tough decision. Should I follow my original plans and go to the lab in Anacostia or should I go to sea? Going to sea had several benefits for me. If I picked the right sort of ship I would be in charge of the electronic equipment and not have a daily ration of orders to follow. I do not do well when ordered, preferring to be asked—a

system which had not yet been discovered by the armed forces. One of the reasons I became an entrepreneur was that I work much better when I am the boss. The Navy had two spots which looked the right size for me—destroyers and submarines. With the food and pay being better on subs, I went that route, being a lover of good food and good pay.

The Navy responded to my request for submarine duty by asking if I preferred Atlantic or Pacific duty. Since my folks lived on the east coast, I opted for the Atlantic, and quickly found myself on a heavy cruiser (the Baltimore) heading for Hawaii. Not long after that I found myself in charge of the electronic equipment on a submarine. In retrospect, I think that I probably couldn't have found anything which I would have liked better.

I spent most of the rest of the war helping to sink Japanese ships (we ran up a very nice score). My sub, the USS Drum, SS-228, is now on display at Mobile, next to the Alabama, where our crew gets together every year for a reunion.

Once the war was polished off I went back to college for more memorization. This was a particularly bitter pill because I'd been through the Navy school and knew how good a school really could be. This whole thing has festered with me over the years, and it came to a head when I began dickering with the presidents of three nearby colleges about the introduction of microcomputer courses. Much to my surprise, I found that these college presidents all agreed with me that memorization is a waste of time.

As I talked with these people, an idea began to develop. What about setting up a college of my own which would concentrate on graduating computerists in a relatively short time? Perhaps an 18-month cram course similar to the one I'd had in the Navy. Students would work ten to twelve hours a day, six days a week, with no vacations, no dances, no soccer teams and no beer busts.

The computer industry is desperate for trained people, as a look at the want ads in the papers will confirm. The *Boston Globe* has virtually a whole section devoted to ads for computer people to help the myriad of firms around Route 128 expand. Wang has set up its own college course to help train people.

By working hard, we could teach the fundamentals of computer hardware, the elements of several languages and concentrate on BASIC and machine language for micros. To round out the education, I would add courses in business such as business law, contracts, personnel selection and management, advertising, sales, marketing, credit, bookkeeping and accounting.

By having the classes followed by lab work—and then by taking on much of the work now being done by the Instant Software people—each student would get a good deal of practical business experience as well as theory. Each graduate would also have a much better resume to offer right out of school by virtue of this practical experience.

The students would be encouraged to write specialized programs, which would then be produced and sold by other students, with royalties to the authors. This income would help pay for the schooling.

The hardware lab would not only teach how to repair all of the micro systems, how to interface anything to anything and how to adjust disks and printers, but would also teach the design of hardware for production by the industry, on a royalty basis.

The idea seems practical, lacking only the management and a campus with buildings. We're working on that aspect of it now, and the prospects seem bright.

How to Become an Expert

One of the best ways to become known in the microcomputer field, and thus share in the juicy consulting fees which are available, is to establish yourself via articles published in *Microcomputing*. It may not exactly be a "publish or perish" situation, but it *is* a "publish or have to work" field.

What should you write about? For starters, why not tackle that aspect of the field which you know best? You'll not only attract more interested readers, but you'll also be on safer ground and save yourself from a lot of embarrassment. Keep in mind that the two major areas for expected development of micro sales lie in the business and educational uses of the systems. Thus, if you have some experience with micros in either of those applications, you may have the makings of an article, or several articles.

There is so much to be written that it is difficult to set up any restrictions. Businessmen are in desperate need of information about what systems, using what software, can solve their problems, cut their costs, increase their efficiency or enable them to provide more and faster service. Educators want to know about the uses of micros in the classrooms, their uses for CAI, the development of student programming, etc. If you've been involved in a project where micros have been brought into a school, there are a lot of people who are interested to learn about your problems and the end results, if any.

On the more technical end, there is a need for articles aimed at newcomers to computing which explain the various types of printers, the many disk systems, the pros and cons of each and how they can be used. Cassette systems should be covered thoroughly, probably in a series. Modems, music accessories, reviews of your experiences with new commercial products that might be useful to others—the list is endless.□

By Robert W. Baker

VIC-20 Debuts

PET/CBM-Compatible Low-Cost Color Computer

Commodore's Low-Cost Color Computer

The VIC-20 (video interface chip) was finally unveiled on January 18 at the Consumer Electronics Show in Las Vegas. It was first demonstrated and released in Japan, where over 1000 orders were taken in the first day of sales. Previously, there were many conflicting rumors concerning the keyboard size, system features and whether or not the unit would be compatible with the current PET/CBM models. Well, good news.

The $299 VIC-20 features a full-sized typewriter keyboard and standard PET BASIC with the familiar screen-editing capabilities. The keyboard even includes eight programmable function keys that can be assigned any BASIC command or instruction set under program control.

System peripherals include a low-priced tape cassette unit, single floppy-disk drive, printer and a broad range of other add-on accessories to custom-tailor the system. I should point out that there is normally no IEEE interface without an added IEEE interface module. The peripherals normally connect through a special serial interface to the VIC-20 (similar to the way Atari interfaces its peripherals), so there will be disks and printers that contain the new interface. You can, however, use an existing PET/CBM cassette tape unit with the VIC-20, or you can use a new lower-cost unit.

The entry-level 5K RAM can be expanded by the use of various RAM (8 or 16K) and/or ROM cartridges to a maximum of 32K. The total add-on memory of any type cannot exceed 27K. Additional special features include an RS-232-C serial interface, joysticks, paddle controllers, light pen, sound and color graphics.

The screen display is normally formatted as a 506-character display, with 23 lines of 22 characters. Graphics resolution is 176 by 176, or 30,976 pixels. Color is selectable for characters, the screen background and border by simple BASIC POKE commands. There are 16 screen colors and eight border colors to choose from in any combination.

Special application programs and games will be available on plug-in ROM form, as well as on tape and disk. Some of these programs will be designed by Commodore programmers, but the majority will come from outside software developers. Commodore is actively seeking good software for the new VIC and has several methods of paying for any software that is used.

Would-be software developers can develop software on the PET/CBM and "down load" to the VIC as long as certain program restrictions are observed. Obviously, no ROM-dependent programs will work, and you should keep in mind the difference in screen size. Commodore will be offering a software development kit that will include necessary hardware and documentation for the development of VIC software in ROM cartridges. I'll keep you posted on any new developments.

The Commodore VIC-20 color computer connects to any television set and comes complete with 5K memory (expandable to 32K), sound, graphics and full-size typewriter keyboard.

More News from Commodore

In light of the tragic fire at the MGM Grand Hotel in Las Vegas during the Comdex 80 Exposition, Commodore has announced the availability of a disaster/emergency plan computer application program. This program will be released to all Commodore dealers. Documentation on Commodore's experiences during their assistance at the Las Vegas fire will also be included. Hopefully, Commodore dealers can then assist all local disaster relief organizations within their area in the event of any future disasters.

Commodore recently announced a new game package titled Galaxy One, which is available on diskette for $49.95. The package includes 24 programs; some previously sold on tape for $8 or more per program. Most of the games are excellent programs with good graphics. Certainly a good bargain for any serious game player. Here's a list of the games that are included:

Lunar Landing	Awari
Jumbo Jet Lander	Wumpus
Concord Lander	Othello
Glider	Super 9 x 9
Spacefight	3D Tic-Tac-Toe
Galaxy Games	Backgammon

Robert W. Baker, 15 Windsor Drive, Atco, NJ 08004.

IF YOU CAN WAIT A MINUTE, WE CAN SAVE YOU $1,000.

With the Starwriter™ Daisy Wheel 25 cps printer from C. Itoh.

A business letter, written on a 45 cps word-processing printer, might take about two minutes to print.

With the Starwriter, it might take closer to three.

The typical 45 cps printer retails for about $3,000.

But the Starwriter 25 retails for about $1,895—thus saving you about $1,000.

And therein lies the biggest difference between the Starwriter 25 and the more expensive, daisy wheel printers.

The Starwriter 25 comes complete and ready-to-use, requiring no changes in hardware or software. It uses industry-standard ribbon cartridges, and it's "plug-in" compatible to interface with a wide variety of systems, to help lower system-integration costs.

Using a 96-character wheel, it produces excellent letter-quality printing on three sharp copies with up to 163 columns, and offers the most precise character-placement available, for outstanding print performance.

C. Itoh's warranty;

3 months on parts and labor, supported by one of the best service organizations in the industry.

$1,000 OFF

Leading Edge Products, Inc.,
225 Turnpike Street,
Canton, Massachusetts 02021

Dear Leading Edge:
I'd like to know more about the Starwriter, and how spending a minute can save me a grand. Please send me the name of my nearest dealer.

Name__________
Title__________
Company__________
Street__________
State__________ Zip__________
Phone: Area Code__________
Number__________

LEADING EDGE. 355

Leading Edge Products, Inc., 225 Turnpike Street, Canton, Massachusetts 02021
Dealers: For immediate delivery from the Leading Edge Inventory Bank™ call toll free 1-800-343-6833
In Massachusetts, call collect (617) 828-8150. Telex 951-624

Reader Service—see page 210

Space Trek	Draw Poker
Polaris	Blackjack
Reverse	Formula 1
Rotate	Wrap Trap
Crypto	Off the Wall
Life	Target Pong

In January Jack Tramiel was appointed Vice Chairman of the Board and Chief Executive Officer for Commodore International Limited. His prior position of President and Chief Operating Officer has been filled by James Finke, formerly Data General Corporation's Vice President and General Manager–Europe. This is the latest in a series of major organizational changes in recent months, as Commodore tries to gain a better position in the American market.


Commodore's newly appointed vice chairman of the board and chief executive officer, Jack Tramiel.

Supersort

Programma International (2908 N. Naomi St., Burbank, CA 91504) is now distributing a machine-language sort program for the PET that was written by Jim Strasma. It's a general-purpose sort utility that takes up less than 1200 bytes at the top of memory. The program works with all current PET BASICs and adjusts to any memory size as it loads. It will not destroy other programs (like DOS 4.0) that have been loaded into high memory. To give you some idea of its speed, Supersort can sort an array of 3000 integers in less than 30 seconds. Try that in BASIC!

You can sort any BASIC string or integer array in ascending or descending order. Arrays can be one- or two-dimensional, and sorts can be specified on any desired field in the second dimension. Additionally, you can subsort other fields when a match is found in the prime sort field. The subsort fields do not have to be in any special order. The fields themselves can be random in length, and no special delimiters are needed. Other options allow sorting by bit mapping, with eight categories per byte. You can even limit sorts to a part of an array if necessary.

The options are set with simple POKE statements prior to calling Supersort via an appropriate SYS command. Convenient default settings are provided for each option. The command area used to pass parameters to Supersort consists of 13 bytes in the second cassette buffer, starting at decimal location 887. This area avoids parts of the buffer used by BASIC 4.0 and DOS 2.1. The command locations are used to indicate the array name, the sort field number, ascending or descending sorts, subsort on matches, bit matching and the matching pattern and/or limited sorts.

During the sort operation, the 44 locations after 886 in the second cassette buffer and the first 44 locations in page zero are exchanged. At the end of the sort, they are restored again. This speeds the sort by allowing shorter machine-language commands using the special zero-page addressing modes of the 6502 microprocessor. It also protects BASIC from the sort and vice versa.

Sorting only part of an array can really speed things up drastically. Typically you dimension a large array, since arrays cannot be redimensioned without losing data. However, much of the array is usually empty or unused. You can tell Supersort where the real data ends and save up to 90 percent of the usual sort time.

With partially filled integer arrays it is essential to use this option because empty integer arrays are filled with zeros. If sorted, the null entries will be placed among the valid zero elements destroying the array validity. In string arrays, the elements of the array are initialized as null strings. Supersort puts null string elements at the top of the sorted array, effectively keeping unused elements away from user data.

The bit matching option takes a little time to understand if you're not used to machine-language programming. It logically ANDs a specified byte within the key field with a selected pattern, which can be any single byte value except zero (1 to 255). If a simple pattern is used many elements can match it. Subsorting can then be used to distinguish these from one another. The primary use of bit matching is to find all data elements that have something in common. You typically define several single-bit flags to identify various global categories or classifications. The bit matching can then be used to find all elements of a particular class or a particular combination of classes.

When errors occur within Supersort, the program sets the normal BASIC status variable (ST) to indicate the detected error. Thus, you can test the value returned in ST to determine if a sort has been accomplished successfully. After an error is indicated, the Supersort command locations in the second cassette buffer are not erased or changed. This allows you to inspect them from the PET monitor, or by peeking from BASIC, to determine the problem. All command locations are normally set to zero after a successful sort. If you should cause an error condition that the program doesn't detect or protect against, the STOP key is still enabled allowing you to terminate execution.

With all the experimenting I've done with this program, I found it to be an excellent utility and apparently "bulletproof." I did not encounter any unexpected or unexplainable situations. Complete documentation, along with a very enlightening demo program, is provided. The program is a worthwhile investment if you do any kind of data manipulations. Why reinvent the wheel, when you have this handy sort program? You can obtain a source listing from Jim Strasma (120 West King St., Decatur, IL 62521) for an extra $10 once you've purchased a copy of Supersort.

Massage and Recover

Last month I reviewed several utility programs from California Software Associates, PO Box 969, Laguna Beach, CA 92652, but only mentioned their Massage and Recover programs. At that time I was not able to load the sample copies due to a bad diskette. I received a replacement disk just after the last column went to print, so I thought I would give more information on these interesting and useful utilities now that I've been able to try them. You may want to refer to the disk format information that appeared in the September 1980 PET-pourri column.

Massage is a disk utility program for the 2000 series PET/CBM systems and disk drives (with the old DOS 1.0 ROMs) written in BASIC. The program can display and edit any diskette block, display and trace sequential or user files, trace block chaining, map the diskette BAM or rename the diskette. When the program is run, a simple menu lets you select the desired function by a single key-letter.

Before giving more information on each function, I should warn that Massage must be used with extreme caution since it allows arbitrary editing of a diskette without safeguards to protect DOS formatting. Edits to the block link pointers, the diskette BAM or the diskette directory could destroy an existing file if not made correctly. The authors suggest that you get used to the program on a scratch disk before trying it on any disk containing valuable information.

However, Massage is quite easy to use, and most mistakes are reversible as long as you can remember what you've changed. A handy appendix is included in the documentation with details on the DOS/diskette format.

EDIT BLOCK (E). When this function is selected, the program prompts for the desired drive, track and sector number. When the drive number is entered, the program accesses the disk and displays the disk name and ID for verification that the correct disk is about to be edited.

When entering the track and sector numbers, only valid combinations are accepted.

Once the necessary information has been entered, the program displays the block link pointers and waits for a RETURN to continue, S to stop or E to edit the line pointers. A track pointer of zero indicates the last block in a chain, and the sector pointer will then indicate the last-used byte of the block.

When you continue, Massage takes a few seconds to read the block, then displays the block contents and asks whether you want to edit, print, stop or go to the next page. The screen display is organized as four pages of the three 20-character columns and one page containing one 16-character column. Each column has a reverse field entry indicating the byte number followed by the decimal value of the byte and the ASCII character it represents (if a printable character). The hard-copy output is organized horizontally for convenience.

If you choose to edit the block, the program first asks for a byte number. Responding with RETURN allows a "multibyte" edit, and you are asked for the starting and ending byte numbers. A maximum of 20 bytes can be edited at one time. After specifying the byte(s) to be edited, you then enter the ASCII character (or string) to be entered.

For nonprinting characters, entering RETURN then allows you to enter the decimal byte value to be entered. If you were performing a multi-byte edit, the same number is repeated throughout the range when a nonprinting character is entered. To help protect against accidentally destroying the disk contents, the edited block is not written to the diskette until you give the STOP command and complete the edit.

DISPLAY FILE (D). This function displays or prints the record number, track, sector and pointer (starting byte) of each record of a selected sequential or user file. Optionally, the record contents can also be displayed or printed. Various controls are provided to suspend or terminate printing. At the end of the file or when told to stop, the program indicates the total number of file records, blocks and bytes up to the point of stopping.

MAP DISK (M). This function displays or prints all directory entries on the diskette. It indicates the filename, file type, total blocks, total bytes, starting track and sector and the directory track, sector and byte. Following the directory information, the diskette block availability map (BAM), number of file blocks, directory blocks, allocated blocks (allocated, but not in sequential or user files) and the total blocks used can also be displayed.

TRACE LINKS (T). This function will trace the block chaining of a selected file or starting track and sector. It displays or prints the indicated links until finding the last block of the chain, and indicates the total number of blocks and bytes in the chain.

UPDATE BAM (U). This function allows you to allocate or free individual sectors on any diskette track. After specifying the desired track, the current BAM is displayed for that track. You can free or allocate any sector in that track, and the updated BAM is displayed for verification.

RENAME DISK (R). This function allows you to change the 16-character diskette name in the disk directory. The current name is displayed prior to asking for the new value. The disk ID is left untouched.

Recover is a BASIC utility program for recovering scratched diskette files. It makes use of the fact that DOS does not change the file contents or chaining when a file is scratched, but instead, frees the file blocks in the BAM and deletes the file-type and starting track codes in the disk directory. Recover tries to reconstruct this information from the file data and the file directory remnants.

If any of the file blocks have been overwritten by another file, then the file can no longer be recovered. Additionally, the wrong information can be accidentally recovered if you select the wrong starting block or another scratched file has "sliced through" the desired file leaving exactly the same number of blocks following the intersection.

After specifying the desired filename, file type and drive number, the program searches the disk directory for any remnants of your file entry. If none are found, the program aborts. Otherwise, Recover checks all possible starting blocks, that is, all blocks with the sector number indicated in the directory. The program checks that the correct number of blocks is in the chain and that each block is not allocated in the BAM.

This process takes some time, but reduces the number of possible starting blocks to a minimum. Remember that the current version of Recover was written for DOS 1.0. When DOS 2.0 is used, the starting track is not changed in the directory when a file is scratched. Thus, a newer version of Recover for DOS 2.0 might be much faster.

After showing the first few bytes of each of the selected blocks, you have the option to stop, accept and reconstruct the file from this block, take a closer look by viewing more data in the block or continue to the next possible block. If and when a starting block is accepted, the program reenters the file type and starting track in the disk directory and reallocates the file blocks in the disk BAM.

Both Massage and Recover are very useful, and possibly very valuable, disk utilities when used correctly. They each perform their respective functions as expected. Sample printouts are included with the printed documentation for Massage.

My only criticism concerns the use of decimal values when displaying various data from the disk. Hexadecimal values may have been much nicer, especially when displaying bit-oriented BAM entries. However, this is really a matter of personal preference and by no means detracts from the overall usefulness of the programs.

As I mentioned *in the last column, the* various utilities can be purchased separately for $60 or as a three-program package for $100. The package price includes Massage, Recover and Edit (reviewed last month). The price of each program might be considered high but they could easily repay themselves many times over in the event of a major disaster. After spending several months creating a new program or entering data in a large data base, what would it be worth to recover all that work if it was accidentally erased?□

By Frank J. Derfler, Jr.

The Six O'Clock Micro News

Brought to You By Atari 800 And the H-89

Two interesting quotes appeared in the news recently. Walter Cronkite, CBS TV's lead newscaster, reportedly said, "It has become impossible to cover the news with the half-hour show. We have a responsibility that we simply cannot discharge." On a tangent subject, Jacques Cousteau is quoted as saying, "The fascinating thing on this planet is that despite all the talk about over-specialization, to a person who's interested, it's much easier to know everything that's going on in the world than it was for Leonardo da Vinci. This is a time of super Leonardos!"

If there is a super Leonardo, and if he or she is looking for more than Mr. Cronkite apparently feels he provides, it is certain data communications will be used to find it. Even if your talents are not up to the level of the Mona Lisa, communicating by computer can cover a wealth of information for you.

Dial-Up Directory is dedicated to the use of microcomputer systems in data communications. In this issue, we will take a look at how two popular microcomputer systems communicate.

The Atari 800

The Atari 800 is a powerful computer system that has been marketed somewhat differently than most. It suffers a little from confusion with the popular video game systems. Atari has vigorously tackled the education market with a well-planned system of educational programs on cassette tapes. The tapes run with a master educational program cartridge. Although Atari has a good disk drive and disk operating system, they have not pushed disk-based business programs.

But, Atari has recognized the data communications revolution and has given the 800 and its less elegant brother, the 400, a good communications capability. Three things are needed to make the Atari into a bright (certainly not dumb, but not too smart either) terminal: Most importantly, a modem is needed to send and receive audio tones over the phone line. The Atari 830 modem is a popular and proven Novation CAT wearing Atari colors. The computers do not have an internal RS-232-C port, so the Atari 850 Interface Module is needed to provide the serial data handling capability. The 850 Interface Module contains a microprocessor and an operating program in ROM. It provides one parallel port and four serial ports.

Finally, you need some software. Atari has a program called TELELINK I which provides an adequate data communications capability. The Atari manual mentions a TELELINK II smart terminal program, but that program does not seem to be available yet. TELELINK I comes in a plug-in program cartridge.

Connecting the Atari system together is easy. The cables and sockets are all high-quality, and the instructions are clear. But the RS-232-C ports do not use the standard DB-25 connectors. Instead, they use a unique socket. Atari says these sockets provide a significant cost and size savings over the DB-25. They say an optional adapter cable is in the works. Fortunately, the pins are clearly defined, and anyone with several different pieces of standard RS-232-C equipment can make an adapter cord, but why should he have to?

The only tricky thing about system set-up is the sequence in which you turn the equipment on. The Atari interface module and disk drive have internal operating programs that have to boot up first so they can answer the computer when it wakes up and asks, "Who's there?" However, the TELELINK I program cannot be used with a disk drive, so the disk must stay off when communicating.

The TELELINK I program becomes operational as soon as the computer is powered up. The program allows the Atari computer to be used as a dumb terminal —when the received or transmitted data scrolls off the screen, it is gone. It also has a unique method of saving received data on a printer connected to the parallel port.

TELELINK I can provide a 1.5K receive buffer in RAM. The program stores received data in this buffer. After 1K of data is stored, the distant end system is given a signal to stop sending (usually a control-S) and the buffer is dumped out to the printer. When the contents of the buffer have been sent to the printer, it is cleared and the distant end is told to start sending again (usually control-Q).

This alternation goes on until all of the desired data is received and printed. In practice, this happens quickly because

The Atari 850 Interface Module provides four serial I/O ports and one parallel I/O port for the Atari 800 and 400 microcomputers. The 830 modem links the system to the telephone lines for data communications.

MICROPROCESSOR SUPPORT I.C.'S

WE GUARANTEE FACTORY PRIME PARTS

We are going to become the largest supplier of prime microprocessor support I.C.'S. We guarantee that our I.C.'S are purchased from manufacturer authorized distributors. This is the only way to deliver prime parts at the lowest possible prices. Our committment is to offer the best price and the fastest delivery to our customer. We give many thanks to our valued customers who have helped us grow.

NEC 16Kx1 DYNAMIC RAM 200 N.S.

These are prime 4116's from one of the best MOS RAM manufacturers in the world.

4116	200ns
8 for $25.00	32 for 96.00

Only prepaid orders on this special.

8080A	CPU	4.95	2708	EPROM	1Kx8	4.95
8085A	CPU	8.95	2716	EPROM	2Kx8	8.95
8086	CPU	99.95	2732	EPROM	4Kx8	21.00
8088	CPU	44.95	4118	STATIC	1Kx8	15.00
Z-80	CPU	10.50	4164	200ns	64Kx1	Call
Z-80A	CPU	12.95	Z80B	CPU		21.00

NEC 1Kx4 STATIC RAM 250 N.S.

These are prime low power static ram's NEC for the finest in MOS MEMORY.

2114L	250ns
8 for $25.00	32 for $96.00

Only prepaid orders on this special.

Z80-P10	7.75	8289	49.95	4050	.69	4531	.99	74C925	6.95	74LS107	.45	74LS244	1.95
Z80A-P10	9.75	4000	.35	4051	1.10	4532	1.25	74LS00	.35	74LS109	.45	74LS245	4.95
Z80-CTC	7.75	4001	.35	4052	1.10	4539	.99	74LS01	.28	74LS112	.49	74LS247	1.10
Z80ACTC	9.75	4002	.35	4053	1.10	4543	1.99	74LS02	.28	74LS122	.55	74LS248	1.10
Z80-DMA	22.25	4006	1.39	4055	3.95	4553	3.50	74LS03	.28	74LS123	1.19	74LS249	1.69
Z80A-DMA	27.75	4007	.29	4056	2.95	4555	.75	74LS04	.39	74LS125	1.35	74LS251	1.79
Z80-S10/0	24.95	4008	1.39	4059	9.95	4556	.75	74LS05	.28	74LS126	.89	74LS253	.98
Z80A-S10/0	29.95	4009	.49	4060	1.39	4581	1.99	74LS08	.39	74LS132	.79	74LS257	.98
Z80-S10/1	24.95	4010	.49	4066	.75	4582	1.01	74LS09	.39	74LS136	.59	74LS258	.98
Z80A-S10/1	29.95	4011	.35	4068	.35	4584	.55	74LS10	.28	74LS138	.89	74LS259	2.95
Z80-S10/2	24.95	4012	.29	4069	.35	4585	.99	74LS11	.39	74LS139	.89	74LS260	.69
Z80A-S10/2	29.95	4013	.49	4070	.49	4702	9.95	74LS12	.39	74LS145	1.25	74LS261	2.49
3205	3.45	4014	1.39	4071	.35	74C00	.39	74LS13	.47	74LS148	1.49	74LS266	.59
3242	10.00	4015	1.15	4072	.35	74C02	.39	74LS14	1.25	74LS151	.79	74LS273	1.75
8155	11.25	4016	.59	4073	.35	74C04	.39	74LS15	.39	74LS153	.79	74LS275	4.40
8185	29.95	4017	1.19	4075	.35	74C08	.49	74LS20	.26	74LS155	1.19	74LS279	.59
8185-2	39.95	4018	.99	4076	1.29	74C10	.49	74LS21	.38	74LS156	.99	74LS283	1.10
8202	45.00	4019	.49	4078	.35	74C14	1.65	74LS22	.38	74LS157	.99	74LS290	1.29
8205	3.45	4020	1.19	4081	.35	74C20	.39	74LS26	.39	74LS158	.75	74LS293	1.95
8212	2.00	4021	1.19	4082	.35	74C30	.39	74LS27	.39	74LS160	.98	74LS295	1.10
8214	3.95	4022	1.15	4085	1.95	74C32	.99	74LS28	.39	74LS161	1.15	74LS298	1.29
8216	1.85	4023	.38	4086	.79	74C42	1.85	74LS30	.26	74LS162	.98	74LS324	1.75
8224	2.65	4024	.79	4093	.99	74C48	2.39	74LS32	.39	74LS163	.98	74LS347	1.95
8226	1.85	4025	.38	4099	2.25	74C73	.85	74LS37	.79	74LS164	1.19	74LS348	1.95
8228	5.00	4026	2.50	4104	1.99	74C74	.85	74LS38	.39	74LS165	.89	74LS352	1.65
8238	5.45	4027	.65	4501	.39	74C85	2.49	74LS42	.79	74LS166	2.49	74LS353	1.65
8243	4.65	4028	.85	4502	1.65	74C89	4.95	74LS47	.79	74LS170	1.99	74LS363	1.49
8251A	5.55	4029	1.29	4503	.69	74C90	1.85	74LS48	.79	74LS173	.89	74LS365	.99
8253	9.85	4030	.45	4505	8.95	74C93	1.85	74LS51	.26	74LS174	.99	74LS366	.99
8255A	5.40	4031	3.25	4506	.75	74C95	1.85	74LS54	.35	74LS175	.99	74LS367	.73
8255A-5	5.40	4032	2.15	4507	.95	74C107	1.19	74LS55	.35	74LS181	2.20	74LS368	.73
8257	9.25	4033	2.15	4508	3.95	74C151	2.49	74LS73	.45	74LS190	1.15	74LS373	2.75
8257-5	9.25	4034	3.25	4510	1.39	74C154	3.50	74LS74	.59	74LS191	1.15	74LS374	2.75
8259A	7.30	4035	.95	4511	1.39	74C157	2.10	74LS75	.68	74LS192	.98	74LS375	.69
8271	60.00	4037	1.95	4512	1.39	74C160	2.39	74LS76	.45	74LS193	.98	74LS377	1.95
8275	32.95	4040	1.29	4514	3.95	74C161	2.30	74LS78	.65	74LS194	1.15	74LS385	1.95
8279	10.80	4041	1.25	4515	3.95	74163	2.39	74LS83	.99	74LS195	.95	74LS386	.65
8279-5	10.80	4042	.95	4516	1.69	74164	2.39	74LS85	1.19	74LS196	.89	74LS390	1.95
8282	6.70	4043	.85	4519	.99	74173	2.59	74LS86	.45	74LS197	.89	74LS393	1.95
8283	6.70	4044	.85	4520	1.39	74174	2.75	74LS90	.75	74LS221	1.49	74LS395	1.70
8284	5.85	4046	1.75	4522	.99	74C175	2.75	74LS92	.75	74LS240	1.95	74LS399	2.95
8286	6.70	4047	1.25	4526	1.15	74C192	2.39	74LS93	.75	74LS241	1.90	74LS424	2.95
8287	6.70	4048	.99	4527	1.75	74C193	2.39	74LS95	.88	74LS242	1.95	74LS668	1.75
8288	25.40	4049	.69	4528	.99	74C195	2.39	74LS96	.98	74LS243	1.95	74LS670	2.29

MAIL ORDERS SHOULD BE SENT TO:
P.O. Box 21432 Seattle, Washington 98111
Telephone Orders & Inquiries (206) 453-0792
Minimum Order $10.00 Add $3.00 Shipping

HANLEY ENGINEERING
RETAIL STORE ✓5
1644 116th NORTHEAST
BELLEVUE, WASHINGTON 98005

FOR THE FINEST IN MICROPROCESSOR SUPPORT I.C.'S

✓Reader Service—see page 210

the parallel printer port is fast, but it does take a finite time which can cost the user money on an information utility system.

This buffer/printer cycle can be placed under manual control, and you can simply ignore the printer and let the buffer overflow. This requires some advanced planning if you want to catch some specific data. You have to clear the buffer so you can catch the new data to dump to the printer. No provision is made within the TELELINK I program for the prestorage of messages for later transmission.

The Atari manual lists several BASIC programs that can be used to transfer data files without using TELELINK I. It also includes an interesting program to allow the Atari to run as a terminal using the five-level Murray, or "Baudot," code. This would be useful for communicating on the various specialized bulletin board systems for the deaf that use this code. The manual that comes with the interface is extensive and detailed. It would be valuable to both the novice and experienced user.

The Atari is a superbly constructed system with great potential. Several things are being done to broaden its appeal. First, a good disk-based word processing program is almost ready for release. Second, Atari is ready with some nice disk-based accounting and inventory control programs. The availability of the popular VisiCalc software by Software Arts, Inc., for the Atari has been a big plus, too. But, a more sophisticated terminal package is needed. This seems to be just a gleam in a programmer's eye at Atari, so there is a lot of room for a small software house to bring out a strong cassette and disk terminal program for the 400 and 800.

By the way, Don Stoner's Microconnection for the Atari eliminates the need for the interface module for modem or RS-232-C operations. It plugs directly into the Atari console in series with the cassette or disk drive and gives both a modem and RS-232-C capability. Don supplies a program that addresses the Microconnection and works in a similar manner to TELELINK. I introduced the Microconnection and other products from the Microperipheral Corporation in the October 1980 column.

Heath/Zenith H-89

The Heath/Zenith H-89 microcomputer is gaining in popularity. It is a high-quality system available at a good price. Kit builders can gain a great deal of knowledge about hardware and save money by assembling the Heathkit version. I've been using Heath equipment since my first AT-1 amateur radio transmitter (how many remember that one?) and have great praise for their kits. A random sample survey of Heath H-89 owners I conducted shows that all were happy with the system (although several were happier when they started using Digital Equipment Corporation's CP/M operating system). Most builders took a mean average of 50 work hours to put their systems together (there were some wild variations).

The H-89 is really a computer and a terminal in one cabinet. The two pieces of equipment communicate using standard RS-232-C signaling. The terminal is completely separate from the computer and can be used as a dumb terminal all by itself. But, why use a dumb terminal when a program like REACH is available?

REACH is a data communications program written and marketed by Walt Bilofsky of Sherman Oaks, CA. Walt runs the Software Toolworks, specializing in H-89 software. He markets many different programs for the H-89 and offers most of them in CP/M versions. REACH runs under the Heath disk operating system, and at $19.95 it is a fantastic bargain.

With REACH, the H-89 becomes a truly smart terminal capable of saving received data and transmitting prepared files. An H-89, at least 16K of RAM and an H88-3 serial interface are needed. Operation is as easy as 1, 2, 3. The H-89 provides a row of special function keys which can be defined by software. REACH comes up ready to communicate. If you want everything displayed on the screen to be simultaneously printed by an attached printer, you push the f_1 function key. If you want received data to be saved in a large buffer, simply push f_2. A special status line on the bottom of the screen shows the option selected and the number of free bytes left in the buffer. This buffer display is a feature many other communications programs could use. A second push on f_2 dumps the buffer to a prenamed disk file. Key f_3 transmits the contents of a file out the RS-232-C port. I don't know of any sophisticated communications program easier to use.

During file transfer, the H-89/REACH system can function in several different modes. The program can simply funnel the data in a steady stream. This would be appropriate for file transfer and for the electronic mail system on The Source information utility. The program can also provide prompted transmission on a line-by-line basis. In this kind of transfer, the host system sends a prompt to the terminal to begin each new line. This prompt is usually the DC1 ASCII character (hex 11), but other characters may be used. REACH can operate in a line-by-line mode with a variety of prompts. CompuServe's EMAIL and several electronic message systems require this prompted mode capability for the transfer of prepared files.

REACH also provides control of auto-answer modems by signaling over certain RS-232-C lines used for special modem functions. An option allows the user to freeze the program and go to the disk operating system without causing the modem to drop the telephone line.

The program comes with a 12-page set of operating instructions. The directions are practical and provide adequate information on operations and various configuration options. The manual assumes the user understands the H-89, its RS-232-C interface, and some ASCII control codes. Inexperienced readers may have to do a little research on some of the terms, but the practical concepts of operation are explained well.

REACH is a bargain. The Software Toolworks has provided a very powerful program at a reasonable price. Their address is 14478 Glorietta Drive, Sherman Oaks, CA 91432.

Help Wanted

If you have any data communications comments or questions, send them to me at the address below. I'm particularly looking for news on amateur radio ASCII and CP/M communications. Any readers outside the US with news and views? Include a stamped envelope or IRC if you want a reply. Send electronic mail to TCB967 on The Source; 70003,455 on CompuServe; or the AMRAD CBBS (703-734-1387).□

Address correspondence to Frank J. Derfler, Jr., PO Box 691, Herndon, VA 22070.

By Eric Maloney

The Language Of Professionals

What Is The Price Of Enrichment?

I'm turning this month's "As the Word Turns" over to guest columnist Robert C. Montgomery. Montgomery is a management consultant, professional corporation director and software author. FINPLAN, his financial modeling system for small business, was published by Hayden Books late last year. He lives at 67 Turtle Back Road West, New Canaan, CT 06840.

"Data are Dead: Long Live Data" (*Microcomputing*, February 1981) is a thoughtful and scholarly comment on the data processing profession's losing battle with the English language. It encourages me to share with others some growing concerns in related areas.

No one can reasonably object to the evolution of the language. It is a continuing, inevitable and desirable process. Were it otherwise, we would still be writing Chaucerian prose, a prospect more alarming than the occasional misuse of a plural noun. But one wants to see improvement of the language, not its debasement by the ignorant. One pays attention when the wonderfully literate editors of the *Wall Street Journal* adopt contractions like "aren't" and "won't." One shudders when a *Microcomputing* author writes "the data is stored."

"Data" is not the only problem. "Criterion" seems to have disappeared, replaced by its plural form. "Criterias" has not yet arisen, but its appearance can only be a matter of time. Infinitives are split like cordwood, and prepositions end sentences as though they were born to the work. "Principal" is confused with "principle," "so" is used as a conjunction, and "nor" has disappeared from English, if not from programming languages. When the industry finally develops computers that can be programmed in English, it will find that programmers don't know how to speak English.

One may argue that a profession has the right to adopt its own usage. Of course. It must certainly be the source of the new words needed to provide for new technology. But the right to adapt existing words and usage should carry with it a responsibility to understand the existing words and usage, and to evaluate the effects of change. Improvements in precision or in conciseness easily justify change, and so can aesthetic values like the way it sounds. But it's hard to accept change based solely upon carelessness, ignorance or —ugly word—arrogance.

There is reason to believe that arrogance is a part of the problem. The data processing profession has enriched the language with such evocative words as "PEEK" and "POKE," and with "bit," "byte" and "nybble." Enriching a language is a far more difficult task than learning it. Surely, then, it is not an inability to learn with which we are dealing, but simply an unwillingness or an ill-conceived belief that the game is not worth the candle.

At least in part, the profession's disinclination to learn the English language is a symptom of contempt for the culture and values of others. There are precedents in other professions. What hospital looks on its patients as customers, deserving of the courtesies that a businessman routinely accords to his patrons? What lawyer takes the time to thoroughly understand his client's business, or values clarity of expression as highly as professional elegance?

Similarly, how many programmers really understand their employers' businesses? How many know what customers want? How many recognize that the results of their work are far more important than the manner in which it is accomplished? How many managers are buying Apples to develop—through their own efforts—the insights which they cannot obtain from their mainframes and the people who run them? Managers are going back to school to learn computers. How many data processing people are going back to school to learn business?

These are not matters of academic interest only. The plain fact is that the data processing professional who hasn't learned English and won't learn about business is hurting himself, his employer and his country.

He is hurting himself by limiting his opportunities for advancement to top management. Today's managers will not promote him far unless they feel that he knows the business and can communicate with others. Little mistakes—in English, as well as in facts—are evidence of carelessness in their eyes, and rightly so. Time won't solve the problem. Tomorrow's managers will be a lot more knowledgeable of data processing, but no more tolerant of fundamental weaknesses.

Data processing professionals with these weaknesses are also hurting their employers, because their inability to understand and to communicate limits the ability of the business to use data processing to its fullest extent, and inhibits the spread of logical and analytical processes to other business functions.

And finally, they are hurting their country because data processing technology will be the key to effective management in the coming years of ever-intensified international competition.

* * * * *

Also writing in response to the February column, Mike Firth of Dallas, TX, observes the birth of a new mutation—"interger."

"I have encountered the use of the word 'interger' in at least three different places recently: A label on a shelf edge in a Dallas computer store; in a full-page ad by Micro-Sci in the February *Byte*, page 135; and repeatedly in the instructions distributed by CompuServe for their Missing Link card and their word processor software.

"There simply is no such word as 'interger.' From the context, it is easy to see that the users mean 'integer,' which is spelled without an 'r' and is related to 'integral' and 'integrate.' It means 'one,' 'to make one,' or 'whole' . . . . The only possible meaning of 'interger' is 'between "gers,"' which is where I think we should leave it." Amen to that.□

We've lowered the

No, the low CompuServe $5.00 per hour charge hasn't changed, but the world of information we offer is growing steadily.

Of CompuServe's 22 large computer systems, 10 are housed here in our Dublin, Ohio Computer Center.

More Computer Power

The number of CompuServe customers keeps growing — now more than 8,000 strong. The more customers we have, the more computer power and access lines we dedicate to our Information Service. Our 22 mainframe computer systems provide sufficient power and capacity to serve thousands more.

More News

In addition to the Associated Press, we've added more news. Read The New York Times and all the news that's fit to print, including Broadway show and movie reviews, stories from Pulitzer Prize-winning writers and more. We are continually adding new features to the information service, so check our "What's New" section regularly to see what other information is available.

TRS-80™ Color Computer and Videotex™

The CompuServe Information Service is sold in Radio Shack® Stores and, in addition to all the major brands of computers and terminals, can be used with the new TRS-80 Color Computer and Videotex terminal. Our special software formats the information in easy-to-read pages. Go forward, backward, anywhere in the CompuServe Information Service databases — even directly to a page of information.

Big System Reliability

When we say we're reliable, we mean it. In fact CompuServe computers were up and running for a 99.6 percent reliability factor last year. We've also built in many "invisible" features like data error detection and retransmission — all those technical enhancements which add up to the CompuServe Information Service being ready to use when you're ready to use it.

By Walter Koetke

Micro Game Strategies

Good Programming, Not Winning or Losing, Is What Counts

Students usually enjoy writing challenging programs, especially when the result can be enjoyed by others. This column will explore three problems of interest to those seeking this type of material. The problems will also be of special interest to those exploring game strategies or artificial intelligence.

Tic-tac-toe

The first example is the traditional game of tic-tac-toe. Of course, you can purchase a tic-tac-toe program from any one of several sources, but that's of little interest beyond learning programming techniques from the program listings. You might also simply enter a program from a published listing such as in the John Kemeny and Tom Kurtz classic, *BASIC Programming*, or in *Creative Computing*'s book *BASIC Computer Games*. Debugging the result as you attempt to find your typing errors can also be instructive. Writing your own program is more instructive than either of the other two options.

All of the commercially available tic-tac-toe programs share a common characteristic: the programmer has fully analyzed the game. As a result, the program always responds to a user's move with the previously determined "best move." With a game such as tic-tac-toe, this type of play will quickly bore any user, as he has no chance of winning.

I suggest writing this program using an algorithm other than one in which all possible board configurations are known. Just to keep you honest, let's consider a three-dimensional game rather than a two-dimensional one. Write a program that plays tic-tac-toe on a $4 \times 4 \times 4$ board, on which you win by placing four markers in a row. What type of strategy should the program attempt? Try having the computer follow the same strategy that will be tried by most opponents.

The majority of users are likely to follow a rather simple strategy. For example:

1. Find three of my markers in a row and place a fourth in that row to win, otherwise
2. Find three of my opponent's markers in a row and place a fourth in that row to block him, otherwise
3. Find two lines of two of my markers in a row so that I can place one marker to have three in a row two different ways, otherwise
4. Find two lines of two of my opponent's markers in a row so that I can place one marker to prevent his trapping me, otherwise
5. Find two of my markers in a row and place a third in that row, otherwise
6. And so forth.

Sounds easy, doesn't it?

Some time ago one of my students wrote a program using a strategy very similar to this one. During the next several months, only one person was able to beat this program.

Those who thrive on game strategy will justifiably be surprised, because an extension of the stated algorithm plays a thorough, uninspired, consistent, but quite beatable game when faced with a forward-thinking opponent. You can, of course, make the program play decidedly better with just a little diagnosis of positional strategy. For example, the eight center positions are of greater value than the eight corners, which are of greater value than the other 48 positions. When placing a marker, considering them in this order will significantly improve the program's playing ability.

Because the final consideration in the strategy list will be making a move at random, the user is likely to see many different sequences of board configurations as he plays several games. A sample program for two-dimensional tic-tac-toe that implements a similar heuristic approach is given in the third edition of *BASIC Programming* by Kemeny and Kurtz. Although Kemeny and Kurtz, the creators of BASIC, wrote this text many years before microcomputers were a reality, the book belongs in the library of every owner or user of a microcomputer.

Writing a program for playing three-dimensional tic-tac-toe that follows a strategy such as this is easier than attempting to anticipate all possible, or even all likely, board configurations. This type of strategy—trying to emulate the play of a user—can also be used in many other games.

When a program that follows this strategy is available, it nicely demonstrates some of those things that computers do very well. Specifically, the computer rapidly follows an explicit set of instructions and does not forget. Certainly, a computer that follows this strategy isn't thinking; it doesn't even know how to play the game better than most users. But the computer doesn't get excited; it always follows the programmed instructions. Nor does the computer overlook any possibilities; its memory is very good.

Hexpawn

The second example is a game called Hexpawn. This interesting little game was first introduced by Martin Gardner in the pages of *Scientific American* (Volume 200, Number 3). The game is played on the familiar 3×3 board of tic-tac-toe. Each of two players controls three pieces, which are initially placed as in Fig. 1a.

The pieces in Hexpawn have moves similar to those of pawns in chess; that is, one space forward to an empty square or one space diagonally to capture and remove an opponent's piece. Players alternate turns until a winner is determined. One wins this game if any one of his pieces reaches the opponent's side of the board or if his opponent cannot complete his turn because he has no legal move available.

Hexpawn is appropriate for introducing the concept of writing a program that "learns" as it plays. The number of different possible games and game situations is very limited; hence a learning program can become very skilled in a hurry. Within eight to ten games, the computer should become unbeatable.

How does the computer learn? I suggest a remarkably simple strategy that determines all moves for a given board

Walter Koetke, Putnam/North Westchester BOCES, Yorktown Heights, NY 10598.

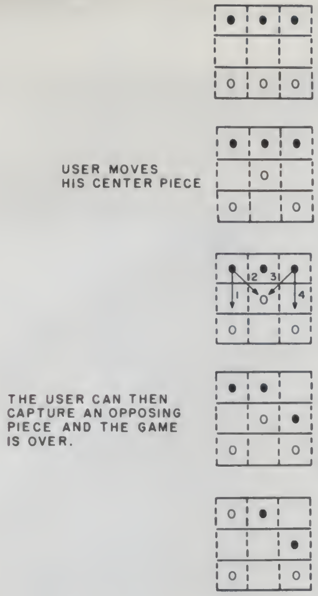


Fig. 1a. *Fig. 1b.* *Fig. 1c.* *Fig. 1d.* *Fig. 1e.*

Fig. 1. Game of Hexpawn.

configuration and then eliminates losing moves. In other words, the computer will *learn to win* only by losing. Nothing is learned by winning. By keeping the strategy so simple, the program does not have to contend with the very clever opponent who purposely loses a few games using a poor strategy in hopes that the computer "learns" some moves as being favorable that really aren't.

Consider the following sequence of play in Figs. 1b–1e, with the computer playing the black pieces. The computer determines that the board configuration in Fig. 1b. has not yet been encountered. Therefore, the configuration is recorded and four possible moves (Fig. 1c) are determined. One of the moves, say number 4, is chosen at random and made (Fig. 1d). The user can then capture an opposing piece and the game is over (Fig. 1e).

After losing the game, the computer should return to the last configuration at which it made a move (Fig. 1c). The program should then record the fact that when this configuration is encountered again, there should only be three possible moves. Because move number 4 led to certain defeat, that option should be eliminated. Thus, when this first game ends, the computer will retain only one board configuration (Fig. 1c) and will have eliminated only one possible move.

Hexpawn makes a nice assembly-language programming exercise. I should underscore, however, that the game's limited parameters make programming in BASIC a very reasonable task.

Finding legitimate, understandable examples of artificial intelligence that can be *readily* programmed in BASIC on a 16K microcomputer is not an easy task. If any readers are aware of other examples that have some of the characteristics of Hexpawn, please send them for possible inclusion in a future column.

Neutron

The third programming example is based on a game introduced by Robert A. Kraus in *Games and Puzzles* (no. 71) and further discussed by Charles Wetherell in *Dr. Dobb's Journal* (no. 35). The game is called Neutron, and I suspect you'll find it considerably more challenging than either of the others already discussed. Neutron is a two-person game played on a 5×5 board. The board with the pieces in their starting position is shown in Fig. 2a.

All moves except the first are composed of two distinct parts. The player must first move the neutron, and then one of his own pieces. Moves are made horizontally, vertically and diagonally. When a move is made, the piece must move as far as possible—until it is adjacent to another piece or against the edge of the board. Pieces are never captured; all 11 are always on the board.

Neutron is won when the neutron piece is moved into your back row by you or your opponent. The game is also won if your opponent cannot complete both parts of his move.

The first player does not move the neutron on his first move. This rule is intended to offset the advantage of having the first move. Figs. 2b–2d are intended to clarify the rules. Clearly the white player's strategy was less than exemplary.

This game can be approached at several different levels:

1. Write a program that makes legal moves selected at random.
2. Develop a strategy that determines the relative value of each move and then proceed accordingly. This illustrates a key aspect of many chess-playing programs in a far less complicated game.
3. Write a program that learns as it plays, as you already did with Hexpawn. Neutron is notably more complex than Hexpawn, and a different sort of learning strategy may be appropriate.

Neutron is definitely a non-trivial game. If you write a program that plays well, please send it to me so it can be shared with other readers.

The programs suggested in this article offer a multi-edged challenge to the programmer. A successful effort requires far more than quickly writing some BASIC code. First, the game must be fully understood. Second, an algorithm for playing must be determined. Third, you must find a method for representing that algorithm on the computer. Fourth, screen formats should be considered so the game is appealing to the user. Finally, the usual steps of program entry, testing and debugging must be completed.

The suggested programs, when finished, will be enjoyed by users and be a feather in their creator's cap. They can also be used as part of a computer literacy program, as each demonstrates a different aspect of strategies used in the study of artificial intelligence.□

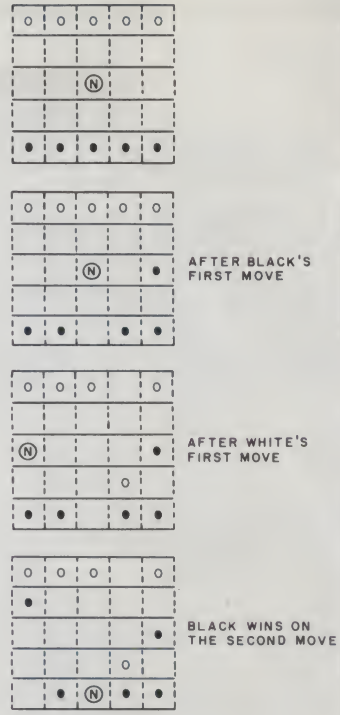


Fig. 2a. *Fig. 2b.* *Fig. 2c.* *Fig. 2d.*

Fig. 2. Game of Neutron.

Breaking the Ice with a Micro
Magazines to Moonlight, Fantasize By
The Red Peril
Micros Underwater

Dynamic Daisywheels

Sales of daisywheel teleprinters could increase by as much as 15 percent by 1985, says a recent report from Venture Development Corp. The study says that the demand for fully-formed character teleprinters will grow with the word processing market.

The study reports that while three U. S. companies control 50 percent of the market for letter-quality printers, certain Japanese manufacturers pose a strong threat. It also says that the future of the daisywheel printer depends on how much of an impact high-resolution dot matrix and ink-jet printers have.

At present, says the report, dot matrix printers can't match the daisywheel for speed, while ink-jet printers are too unreliable. Both are still in the research and development stage.

For more information on the report, which is titled *The Teleprinter Industry: A Strategic Analysis*, contact Wendy Abramowitz, Market Research Analyst, Venture Development Corp., 1 Washington St., Wellesley, MA 02181 (617-237-5080).

The Ultimate Icebreaker

Having trouble getting people at your parties or conventions to interface with one another? Do your computer buddies become catatonic when they're dragged out of their hovels and forced to face humanoid life forms?

Arizona Digital Corp. thinks it has the right product for you—the Alter Ego.

ADC has packed the punch of a values clarification workshop into a microcomputer the size of a cigarette lighter. Any two people with Alter Egos can immediately tell whether they're compatible, no conversation or other human interaction necessary. Dr. Joyce Brothers calls it "the ultimate icebreaker."

Before the function, each person enters the answers to 96 questions into his Alter Ego. The people then walk around and touch Alter Egos, receiving a digital reading that supposedly indicates levels of agreement and compatibility. They can then shake hands or throw punches, depending on how compatible they are.

The questions come in two booklets—Personality Factors and Management Style. The former includes queries on such topics as sexual drive, relationships, independence, and one's sense of responsibility. The latter asks about such matters as interaction with other employees, work-related character traits, communication, career goals and company management.

ADC says that the devices let people meet without exposing their inner feelings. They also make conversation pieces, whether the digital reading indicates compatibility or incompatibility.

The Alter Ego costs $29.95, with the per-item price decreasing according to quantity. For more information, write Arizona Digital Corp., 4246 E. Wood St., #330, Phoenix, AZ 85040.

Mag for Moonlighters

There are lots of ways to make money with a microcomputer. One of them is to *tell* people how to make money with a microcomputer. That's what J. Norman Goode is doing, with his newsletter *Micro Moonlighter*.

The monthly periodical, says Goode, is for the person "who has a system at home and wants to recoup his investment or use his micro for a cottage industry."

The newsletter will include "quick hints that the reader can use to improve his business," Goode says. He plans to run several regular columns, including one on advertising and one on computers and the law. Goode plans for the moment to avoid feature-length articles, concentrating instead on short tidbits of usable information.

Goode is convinced that there is a market for his newsletter. "Everyone's coming out with one or more books on how to make money with a micro," he says. "Obviously, if the book publishers are releasing these titles, someone must be buying them."

For the moment, Goode is comfortable with the newsletter format. Although *MM* might eventually metamorphose into a magazine, Goode doesn't want it to become too slick.

"I would like to eventually solicit articles and be able to pay for them," he says.

Micro Moonlighter is scheduled to premier in May, and will run eight pages. Charter subscriptions are $25 a year in the U. S., $29 in Canada and $35 worldwide.

For more information, contact Goode at 2115 Bernard Ave., Nashville, TN 37212.

CBM Puts Together Disaster Program

Commodore Business Machines has put together a disaster/emergency plan computer program to be distributed nationally to some 500 dealers. The program, which was a direct result of the MGM Grand Hotel fire in Las Vegas last November, is designed to help local relief organizations in the event of a disaster.

CBM mobilized seven business systems after the MGM fire, to help keep track of the whereabouts of the thousands of people in the hotel at the time. The computers were also used to help people make air and ground transportation arrangements.

The program was designed by Brian

Padol of Micro Research, and is based on a Commodore information list management system program.

"We just came to the realization that most large municipalities are not equipped to handle such an emergency," says Margaret Jillett, director of public relations at Commodore.

Magnetic Fantasies

If your appetite for computer fantasy games and related material is insatiable, you might want to try a new bimonthly magazine called *Magnetic Fantasies*.

It costs $2.95, which is a lot of money to pay for only 28 pages. Furthermore, *MF* is sloppily put together, and the writing is generally atrocious. But the editors partly redeem themselves with good spirits and a freewheeling sense of humor. From the Tablet of Contents to the liberal use of parenthetical witticisms from the editors, *MF* has a hard time taking itself or its topic very seriously.

The first issue came out in February, and is highlighted by an article titled "On the Lack of Morality in Adventuring." Author Clyde Primm, an obscure L. A. actor, bemoans the fact that Adventure responds to such commands as fight, attack and kill but not to love, caress, calm and soothe.

"Morality must be brought back into this game," he says. "Programmers must allow for statements like, 'Soothe dwarf,' 'Coo to bird,' 'Ask mummy what he has been doing for the last five centuries.' This would stop the corrupting influence this game has on our youth."

The first issue also includes a bibliography of fantasy game programs appearing in the major microcomputing magazines, a bibliography of articles about fantasy games, a list of the games currently on the market and several software reviews.

Editor Richard Koch says that *MF* had an initial press run of 500 copies, 100 of which have been distributed through an announcement on several computer bulletin board services. The response, he says, has been good, and he hopes that the magazine can grow into a full-blown enterprise.

But if *Magnetic Fantasies* is serious about its future, it will have to improve its writing and editing. A sense of humor is not enough to carry a magazine to the Magic Kingdom of Success.

For more information, write the editor at 100 S. Edgemont, Apt. 1, Los Angeles, CA 90004.

The Red Peril

Is *Kilobaud Microcomputing* inadvertently aiding and abetting a Communist conspiracy to steal high-tech secrets from the U. S.? Apparently so, according to right-wing columnist Phyllis Schlafly.

"The Soviet KGB has been thorough,

ingenious and daring in its operation designed to steal American computer technology," she says. "For starters, the Soviets simply bought U. S. technical magazines from bookstores and borrowed them from libraries."

In addition, she says, the Soviets have been buying up millions of dollars' worth of microcomputer equipment and illicitly shipping it back to the Old Country.

The Institute of Strategic Trade, she further claims, has gathered "a hair-raising collection of tales of espionage, large cash payments and businessmen making tremendous profits on illegal deals while paying insignificant fines."

In view of this frightful situation, we suggest that you memorize this magazine and then eat it.

No Clock for Apple III

The Apple III microcomputer will no longer include its special built-in clock/calendar circuit. The price of the Apple III has thus been reduced by $50.

The company says that it has not been able to find a satisfactory supplier.

.The battery-integrated circuit is used to log time and date information automatically on files the computer has stored. Users can enter this information manually on the keyboard.

Customers with Apple IIIs will receive a $50 rebate. Anyone who sent in their warranty card should get a letter and rebate offer.

Diving into Computers

Microcomputing's assistant managing editor Sue Gross was out in Las Vegas recently, and brought back this report:

Practically next door to the Consumer Electronics Show, the Diving Equipment Manufacturers Association was at the Hilton Convention Center with its own exhibition. Among the hundreds of exhibitors, I found three companies using microcomputers.

AquaComp was showing a complete microcomputer system for dive shops. Sherwood was displaying their regulator evaluation program for the Apple II. And Jeppesen/Sanderson was showing how they are developing a program for diving instruction that includes a microcomputer, video tapes and video disks.

AquaComp's system, which is also called AquaComp, uses OSI's C2 4P, with disk drives, CRT and printer. The system covers every aspect of the diving business, and includes such programs as Instruction, Travel, Student File (see Fig. 1), Repair and Parts Inventory and Rentals and Rental Inventory. It also includes such general programs as Payables and Receivables, Inventory, Purchasing and General Ledger. As an option, AquaComp can be made into an electronic cash register with a cash drawer. The price is $7000.

AquaComp has a password security system, as well as a built-in patented ROM chip specific to each machine. Thus, the software cannot be run on any

```
ENTRY# 5
NAME                              JERRY M. KATZ
STREET ADDRESS                    567 EAST 18TH COURT
CITY                              ST. PAUL
STATE                             MN
ZIP CODE                          12345
HOME PHONE NUMBER                 (123) 456-7890
OCCUPATION                        PROFESSOR
EMPLOYER                          UNIV. OF MINN.
SEX (M or F)                      M
DATE OF BIRTH (mo/day/yr)         8/4/36
HEIGHT (inches)                   68
WEIGHT (pounds)                   155
HAIR COLOR                        BROWN
EYE COLOR                         BLUE
IF PHYSICAL TAKEN (mo/day/yr)     2/16/81
PHYSICAL GIVEN BY DR.             CHARLES O. TUCEK
PHYSICAL CONDITION (comments)     GOOD
DATE CLASS BEGINS (mo/day/yr)     3/19/81
CLASS LEVEL                       BASIC
CLASS LOCATION                    PARADISE HEALTH CLUB
NAME OF INSTRUCTOR                BOB ROBERTS
TOTAL COURSE PRICE ($)            150.00
PAYMENT ARRANGEMENTS              DEP. $50--BAL. 1ST CLASS
TOTAL PAID TO DATE ($)            $50.00
LAST PAYMENT MADE (mo/day/yr)     1/25/81
NEXT PAYMENT DUE (mo/day/yr)      3/19/81
PERFORMANCE ON POOL TEST          AVERAGE
NUMBER OF CLASS SESSIONS          10
NUMBER OF SESSIONS ATTENDED       8
FINAL WRITTEN EXAM SCORE          100
FINAL POOL TEST SCORE             100
CLASS WORK (comments)             EXCELLENT
POOL SKILLS (comments)            ADEQUATE IN SWIM TEST
NUMBER OF CHECK-OUT DIVES         3
CHECK-OUT SKILLS (comments)       1ST DIVE-TROUBLE W/MASK
CERTIFICATION ISSUED (Y or N)     Y
CERTIFICATION DATE (mo/day/yr)    6/17/81
CERTIFICATION AGENCY AND LEVEL    PADI-BASIC
CERTIFYING INSTRUCTOR             BOB ROBERTS
TOTAL EQUIPMENT PURCHASES ($)     249
LAST EQUIP. PURCHASE (mo/day/yr)  1/16/81
TOTAL TRAVEL PURCHASES ($)        0
LAST TRAVEL PURCHASE (mo/day/yr)  0
RATED CLASS--1=EXCLNT/5=POOR      3
```

Fig. 1. Typical AquaComp student file.

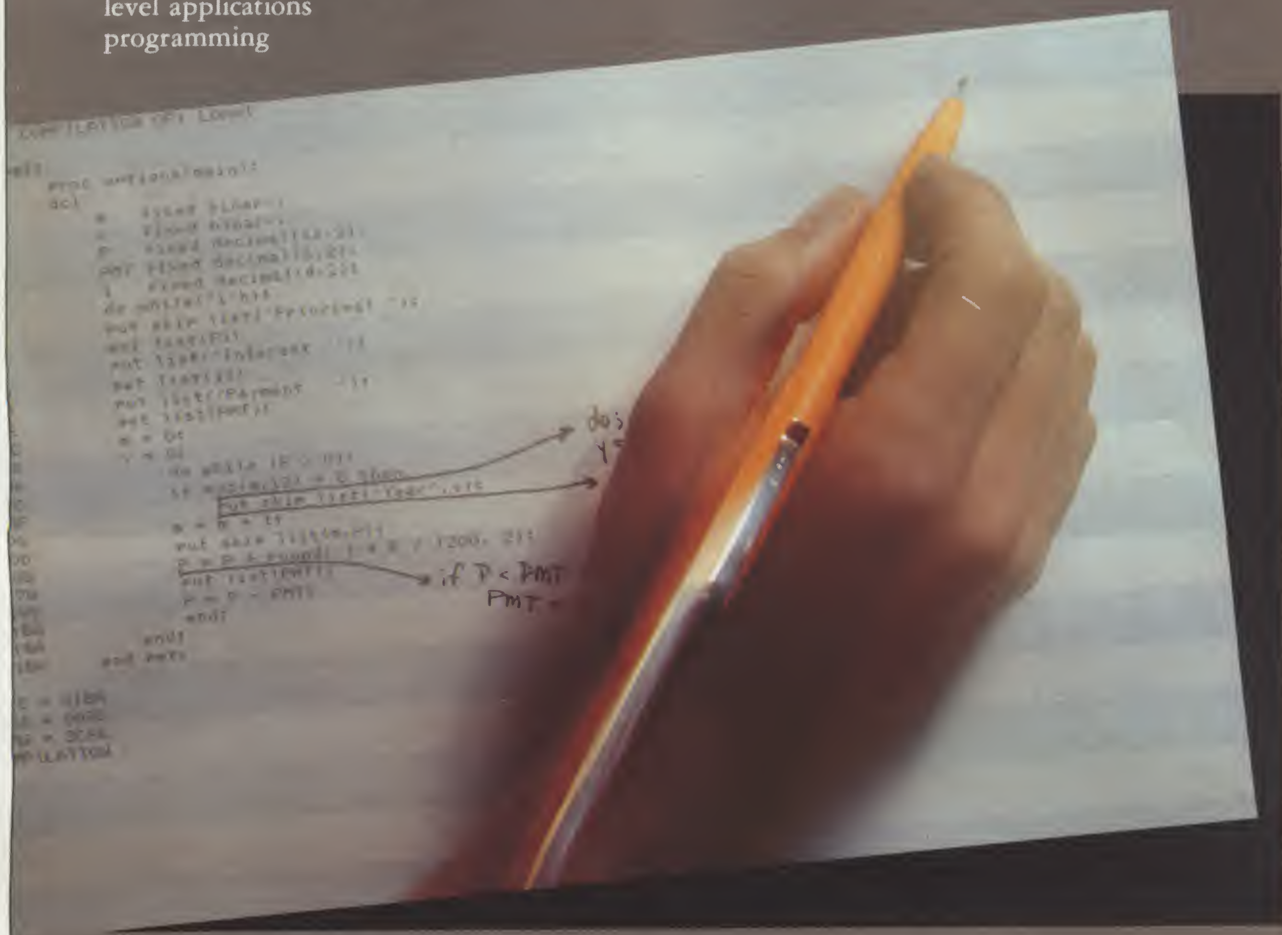


Photo 1. Sherwood's Apple II in use with the hyperbaric chamber.

other microcomputer. The programs can be copied for backup purposes with a copying program supplied with the system.

When I was at AquaComp's exhibit, they already had 50 committals. Not bad for a few days' work.

AquaComp is a division of American Research Service Corp., 175 Fifth Ave., No. 1101, New York, NY 10010.

At the large Sherwood Selpac booth, I found an Apple II being put to an industrial use. Sherwood (120 Church St., Lockport, NY 14094) makes valves and regulators for scuba diving. They are using their Apple II with a hyperbaric chamber, a breathing machine and a pressure transducer to test their regulators for efficiency.

In essence, the Apple II receives signals through an A/D converter from the pressure transducer, which is placed inside the regulator being tested. The regulator is hooked to the breathing machine inside the hyperbaric chamber, which simulates the water pressures experienced when scuba diving.

The program, written by Robert J. Miller of Freeville, NY, first dimensions the pressure data arrays. It then clears the screen and displays the menu of options (a. calibrate the pressure transducer; b. run a test; c. end session). After the pressure transducer is adjusted, tests can be run, and the operator then enters pertinent data about the test.

The breathing machine has a switch to trigger the data gathering phase at the start of the breathing cycle. The Apple II loops through the data arrays and stores the raw pressure data (up to 400 entries). This data is examined and run through formulas to calculate kilogram-meters per liter, and another array is filled with the modified data. A grid is plotted with x-y coordinate values and the test criteria are printed at the bottom of the screen (see Photo 1). The smaller the size of the pressure volume loop, the less work the diver's lungs must do and the more efficient the regulator.

This program and the Apple II let Sherwood make fast, accurate tests of their own, and their competitors', regulators.

Their Apple II, by the way, is just a 16K model, and uses only a cassette.

The Jeppesen/Sanderson booth was demonstrating the use of video tape for diving instruction. Upon further discussion with R. D. Swanson, director of marketing for special programs, I learned that Jeppesen/Sanderson has been developing interactive teaching systems incorporating a computer and a video disk.

The diving program is still in its early stages, but J/S has been selling a similar program to flight schools to help with judgmental training. They are also working on a program to train bank tellers how to spot fraudulent checks, hot checks, counterfeit money, and so on.

The beauty of this type of system is the video disk's random access feature. It's ideal for use with a computer in interactive training, because the computer can go back and cover points that the student has missed or needs more help on.

As a final note, the DEMA show wasn't the only place computers were in evidence. Not having been in Vegas for many years, I was surprised (I suppose I should have expected it) to see the profligate abundance of computerized one-armed bandits (Bally slot machines). Pull the arm, and the beautiful color graphics spin just like the real thing. There are various computer card games too. Naturally, I lost.

MICRO QUIZ

What Does This Program Do?

```
10  A=1 : B=1 : S=0
20  FOR I=1 TO N
30    S=S+A
40    C=A+B
50    A=B
60    B=C
70  NEXT I
80  END
```

In N=9, what is the value of S after the program is executed?

(answer on page 192)

The first personal computer for under $200.

The Sinclair ZX80.
A complete computer— only $199.95 *plus $5.00 shipping.*

Now, for just $199.95, you can get a complete, powerful, full-function computer, matching or surpassing other personal computers costing several times more.

It's the Sinclair ZX80. The computer that "Personal Computer World" gave 5 stars for 'excellent value.'

The ZX80 cuts away computer jargon and mystique. It takes you straight into BASIC, the most common, easy-to-use computer language.

You simply take it out of the box, connect it to your TV, and turn it on. And if you want, you can use an ordinary cassette recorder to store programs. With the manual in your hand, you'll be running programs in an hour. Within a week, you'll be writing complex programs with confidence.

All for under $200.

Sophisticated design makes the ZX80 easy to learn, easy to use.

We've packed the conventional computer onto fewer, more powerful LSI chips—including the Z80A microprocessor, the faster version of the famous Z80. This makes the ZX80 the world's first truly portable computer (6½" x 8½" x 1½" and a mere 12 oz.). The ZX80 also features a touch sensitive, wipe-clean keyboard and a 32-character by 24-line display.

Yet, with all this power, the ZX80 is easy to use, even for beginners.

Your course in computing.

The ZX80 comes complete with its own 128-page guide to computing. The manual is perfect for both novice and expert. For every chapter of theory, there's a chapter of practice. So you learn by doing—not just by reading. It makes learning easy, exciting and enjoyable.

You'll also receive a catalog packed with items that can make your ZX80 even more useful. Including 27 program cassettes, from games and home budgeting for just $6.95, to Sinclair's unique Computer Learning Lab (a workbook, six cassettes with 100 lessons, and two cassettes for storing programs).

ZX80's advanced design features.

Sinclair's 4K integer BASIC has performance features you'd expect only on *much* larger and more expensive computers.

- *Unique 'one touch'* entry. Key words (RUN, PRINT, LIST, etc.) have their own single-key entry to reduce typing and save memory space.
- Automatic error detection. A cursor identifies errors immediately to prevent entering programs with faults.
- Powerful text editing facilities.
- Also programmable in machine code.
- Excellent string handling capability—up to 26 string variables of any length.
- Graphics, with 22 standard symbols.
- Built-in random number generator for games and simulations.

Sinclair's BASIC places no arbitrary restrictions on you—with many other flexible features, such as variable names of any length.

And the computer that can do so much for you now will do even more in the future. Options will include expansion of 1K user memory to 16K, a plug-in 8K floating-point BASIC chip, applications software, and other peripherals.

Order your ZX80 now!

The ZX80 is available only by mail from Sinclair, a leading manufacturer of consumer electronics worldwide.

To order by mail, use the coupon below. But for fastest delivery, order by phone and charge to your Master Charge or VISA. The ZX80 is backed by a 10-day money-back guarantee, and a 90-day limited warranty which can be extended by 12 months under Sinclair's extended service program for $25.00.

Price includes TV and cassette connectors, AC adaptor, and 128-page manual.

All you need to use your ZX80 is a standard TV (color or black and white). The ZX80 comes complete with connectors that easily hook up to the antenna terminals of your TV. Also included is a connector for a portable cassette recorder, if you choose to store programs. (You use an ordinary blank cassette.)

The ZX80 is a family learning aid. Children 10 and above will quickly understand the principles of computing—and have fun learning.

To order call toll free: 800-543-3000.
In Ohio call: 800-582-1364.
Ask for operator #508.
Phones open 24 hours a day, 7 days a week.

sinclair **Sinclair Research Ltd., One Sinclair Plaza, Nashua, NH 03061.**

To: **Sinclair Research Ltd., One Sinclair Plaza, Nashua, NH 03061.**
Please send me ______ ZX80 personal computer(s) at $199.95 each (US dollars), plus $5 shipping. (Your ZX80 may be tax deductible.)
Send me ______ Computer Learning Lab(s) at $49.95 each.
Register me for ______ extended service program(s) at $25.00 each.
I enclose a check/money order payable to Sinclair Research Ltd. for $______.

Name ______
Address ______
City ______ State ______ Zip ______
Occupation ______ Age ______
Intended use of ZX80 ______
Have you ever used a computer? ☐ Yes ☐ No Do you own another personal computer? ☐ Yes ☐ No

0-4-KM

Use It with TRS-80™ Computer Models I, II, and III

Designed for the Decision Maker, VisiCalc is so effective that if no other computer program existed, VisiCalc alone would justify your owning a TRS-80. It's an all-purpose planning and modeling program that dramatically changes the way you make calculations. Ideal for businessmen, scientists, accountants, engineers, chemists — anyone wanting to solve, examine and revise complex formula-oriented computations at the touch of a button.

The "Electronic Worksheet." VisiCalc adapts itself easily to your specific numerical problems, big or small. The computer screen becomes a window through which you view a section of an electronic worksheet. This sheet can be divided into as many as 63 columns and 254 rows (there are some restrictions depending on amount of available memory). In each "entry position" you can insert a description, number, or formula for computation. VisiCalc then automatically performs all calculations, displaying the results instantly.

Here's Where the Magic Begins. Suppose you have a set of complex design equations. VisiCalc lets you define and store them and see the effect of changing any variable. With every change VisiCalc instantly recalculates all affected parameters. Built-in transcendental and statistical functions make VisiCalc perfect for a wide range of engineering and scientific applications. You get the speed, storage and printout capabilities of the computer — without the programming work.

Amazing Versatility. VisiCalc is a powerful tool for everything from corporate financial planning to engineering analysis to family budgeting. Sales projections, cost estimation, and thousands of other time-consuming chores can be done in minutes — more thoroughly than ever before! What's more, you can easily print specified portions of your worksheet with an optional printer or store it all on disk for later use.

And It's So Affordable. VisiCalc software is available in three versions: 32K disk Model I and 32K disk Model III ($99.95 each) and 64K Model II ($299). You can get your system started for only $2495 with a 2-disk Model III, plus software, then add a printer later for as little as $799. All available at Radio Shack's 6100 stores and dealers and 123 Computer Centers, and supported by 135 Service Centers, nationwide. Leasing and credit plans available. Come see what VisiCalc can do for you today!

Radio Shack®

The biggest name in little computers™

Retail prices may vary at individual stores and dealers. Special order may be required initially.

™VisiCalc is a Trademark of Personal Software, Inc.

Send For More Information on VisiCalc or TRS-80 Computers

Radio Shack, Dept. 81-A-249
1300 One Tandy Center
Fort Worth, Texas 76102

NAME ____________________

STREET ____________________

CITY ____________ STATE ________

ZIP ____________ PHONE ________

RADIO SHACK DIVISION OF TANDY CORPORATION • 6100 STORES AND DEALERS, 123 COMPUTER CENTERS AND 135 SERVICE CENTERS NATIONWIDE

Revealing Secret Code

I particularly enjoyed Alan Sclawy's excellent article in your November issue, "CP/M Encryption Prescription" (p. 42).

As the author pointed out, one defect in his program is that plaintext characters must be inserted if the encryption process results in certain control codes. A solution is to map the control codes onto the ASCII codes from 128 to 159.

At the end of the encryption cycle, test the resulting code to see if it is less than 32; if so, complement it with "NOT"; reset the sign bit to zero with "XOR – 128"; and add 32. The process is simply reversed at the beginning of the encryption loop.

In CBASIC, we have the following at the start of the loop:

```
IF P%>127 THEN P% = NOT(P% – 32) XOR ( – 128)
```

At the end of the loop is:

```
IF P%<32 THEN P% = ((NOT P%) XOR ( – 128)) + 32
```

I think you are performing a public service by helping to disseminate knowledge of cryptographic techniques, and I hope you publish more articles on this subject.

John A. Thomas
Dallas, TX

See p. 64—Editors.

Megabytes on Cassette

A product has been on the market now for about a year that does just what you asked for in your November 1980 "Publisher's Remarks" (New Ideas, p. 8)—that is, allow the storage of ten megabytes of Winchester data on a regular cassette.

The Corvus Mirror is a low-cost backup for the Corvus Winchester disk system employing standard video recording technology. It converts the output of the Corvus 10 or 20 megabyte Winchester disk into a stream of serial bytes which can be recorded on any videotape recorder. The most common video machine used for this purpose is the video cassette recorder because it offers the convenience and low cost of easily available cassettes.

Data is backed up at a one megabyte per minute rate, permitting the entire contents of a Corvus Winchester disk to be copied in ten or 20 minutes. Total capacity of a cassette is 120 megabytes. Thus, the Mirror serves not only as a security backup, but also provides archival storage. In a larger sense, it allows microcomputer users to enjoy both the sealed environment reliability of the Winchester disk system and the convenience of cassette loading at the same time by reloading cassette data back onto the disk as needed.

Joseph D. Hughes
Vice-president, Marketing
Corvus Systems, Inc.
San Jose, CA

All Shapes and Sizes

Arnold Bragg's article "Area Estimation" (October 1980, p. 112) showed an amazingly effective algorithm for its simplicity. Its greatest weakness is that only straight lines (not curves) are fitted between the data points.

To test the formula for its ability to fit curves, I used a circle with the radius of 1 and chose the data points so they were equally spaced around the circumference. For comparison, I also calculated the area of an N-sided inscribed polygon. The area determined by the formula is always the area of the inscribed polygon.

Fitting a curved line is more of an inconvenience rather than a fault of the algorithm. Any inconvenience the fitting creates is more than made up for by the algorithm's ability to fit very complex shapes. The only time the algorithm failed was for the ring figure, and I tried to do the outside and the inside area in one pass. This could easily be handled by making two passes—one for the outside and the second one for the inside—and then subtracting the differences.

Bob Fruit
Hinsdale, IL

A Change in the Standings

Here's some new information that changes some of the data in your recent article "The 16-Bit Time Trials" (*Kilobaud Microcomputing*, October 1980, p. 182).

The Z8000 CPUs are now available from both Zilog and Advanced Micro Devices in 6 MHz clock rate versions (called the 8001A and 8002A). Use of the A version reduces cycle times to two-thirds that of the 4 MHz parts, resulting in execution times and rankings in your Table 2 of 32 us (#1) for table lookup, 712 us (#3) for block move, 5.3 us (#1 tie) for jump table and 13.3 us (#3) for multiply. The execution time index changes to 0.596 (#2), and the overall index changes to 0.799, which makes the Z8000 number one in rank.

John Springer
Advanced Micro Devices, Inc.
Sunnyvale, CA

In Praise of Kit Building

We enjoyed very much your January feature on "Assemble a Super Business System" by Dr. Tom Lukers. It certainly highlighted the money savings and expanded design capabilities one can achieve from building his own computer system in kit form.

Congratulations to you and Dr. Lukers on a fine article.

Myron Kukla
Advertising and Public
Relations Coordinator
Heath Company
Benton Harbor, MI

The Importance of Being Zero

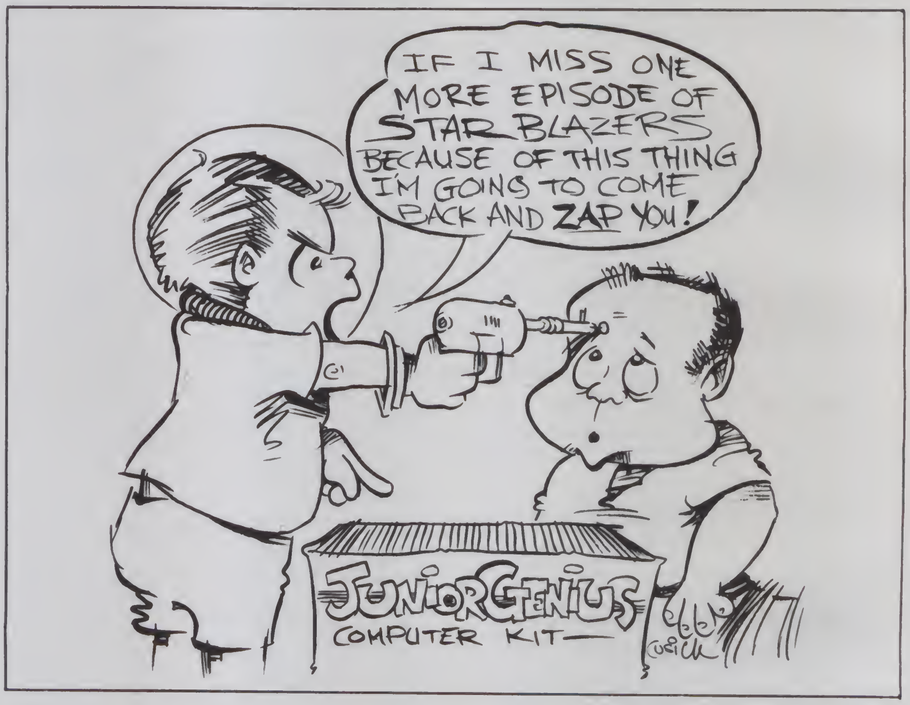
Regarding Messrs. Price and Piescik's disagreement about zero numbering (Letters to the Editor, p. 27, October 1980), zero numbering is definitely *not* "an established design practice that novices might as well go along with." Zero numbering will seem rather pointless until one enters that "mysterious" world of assembly language. Then, zero numbering becomes a necessity and not some mere tradition.

First is the ease of indexing. In Mr. Price's example of the tracks on a floppy disk starting at track 0, suppose our program inputs any track number and looks in a table called DDATA which contains one byte per track to tell us what is on that track. Now remember that the label DDATA is actually the address of the beginning of the table, which is the first entry. The track number is merely added to the address DDATA to obtain the correct byte. That is, to reference the first track (track 0), zero is added to DDATA, and since DDATA points to the first entry, the correct offset is chosen. If the first track was track #1 instead of #0, then either one would have to be subtracted from the track number before indexing, or the first entry in DDATA would have to be filled

(continued on page 211)

FCC Takes Aim Against RFI Polluters

By Chris Brown and Eric Maloney



Deep within the confines of the CPU, hidden away in an orderly world of LSI clusters, crisp square waves obediently cycle and time, control and consort. Strict schedules keep them moving in lockstep as data is fetched, registers are incremented, stack pointers set and numbers crunched. Speed is of the essence: the user is an impatient master hoping to save a microsecond here, a nanosecond there. Frequencies run to the megahertz, and the benefits of brevity in instruction implementation are paramount.

But feeding on the square waves' speed and strength is the bugaboo of the computer revolution—radio frequency interference. Disruptive and counterproductive, RFI takes the form of thousands of harmonically related multiples of the square wave's fundamental frequency.

This RFI renders nearby communications receivers unusable. Just sit in a room with a working microcomputer, and listen to a radio. It doesn't matter what frequency you tune to—easy listening FM, Top 40 AM, air traffic control, amateur or emergency. As the machine cycles through its instruction set, there is intermittent noise from 1 to 1000 MHz, an annoying staccato buzz coming from the processor through the keyboard, along the spaghetti maze of wiring, over the recorder leads, out of the ribbon cables and even through the ac power line. The hash pervades the surrounding environment.

It was only a matter of time before RFI caught the attention of the Federal Communications Commission, the government agency responsible for refereeing the use of the radio spectrum. After three years of study, the FCC handed the microcomputer industry a mandate: Get rid of the RFI by January 1981, or close up shop.

The FCC wasn't kidding—the new rules are tough. Microcomputers marketed for home use (called class B computing devices) may not generate appreciable RFI from 450 kHz (just below AM radio) to 1000 MHz (above UHF-TV). The same regulations apply to peripheral devices, whether sold separately or as part of a system.

Microcomputer manufacturers must also follow a complex, time-consuming and expensive certification procedure. A note from Mom simply won't do—the manufacturer has to test its product, submit the test information to the FCC, complete application forms, submit photographs and pay certification fees.

The price for noncompliance is harsh. The equipment cannot be marketed or advertised in the U.S., whether manufactured domestically or imported.

By last fall, it became obvious to many companies that they could not meet the January deadline. Firms like Apple, Commodore, Exatron, Lobo and Heath went to the FCC praying for mercy and stalling for time. They all won extensions to April 1, but the reprieve did little to relieve the pres-

Chris Brown is the assistant technical editor of 80 Microcomputing, *and Eric Maloney is on the* Kilobaud Microcomputing *staff.*

sure. In fact, as of press time, a few observers speculated that some firms might have to halt production, or perhaps even go out of business, if further extensions were not forthcoming.

For most manufacturers, the cost will be high. The final bill for personnel, equipment, testing and modifications could run into the hundreds of thousands of dollars.

And what will be the impact on the consumer? Most obviously, he will have pay the costs through higher prices. He also might find it difficult to interface certain systems and peripherals. And, in the long run, he might find fewer new products on the market.

Photo 1. *Interior of the RF-proof, shielded chamber of R and B Enterprises' test facility. A PET micro is being tested. Photos and figures courtesy of R and B Enterprises.*

Inevitable

Considering the history of RFI regulation, the new rules concerning micros come as no surprise. From the early days of diathermy to present-day data processing, the FCC has moved slowly but effectively to limit the interference potential of many types of devices. In 1947, an adolescent FCC allocated certain frequencies for use by industrial, commercial and medical equipment with the provision that these devices meet technical standards that specified limitations on radiation. More recently, the FCC moved to control RFI-generating CB sets.

"Because the spectrum is a valuable but limited resource that can be used in various but incompatible ways, simple economic efficiency suggests that such a resource be employed in its most valuable way," says a Microcomputer Industry Trade Association (MITA) report. "By issuing its new rules the FCC has decided that radiation from electronic devices is a less valuable use of the spectrum than the radio services with which they might interfere."

The new rules distinguish between two kinds of micros: those marketed for use in commercial settings (class A) and those intended for use in residential settings (class B). In each case, the marketing strategy of the manufacturer determines the class of the device. Class B computing devices include the Heath H8 and H-89; the TRS-80 III, and Color Computer; the Apple II; the Commodore PET; and OSI C1P and C4P; and the Atari 400 and 800.

The class B technical standards are much tougher than those of class A (see Figs. 2a and 2b). The FCC's reasoning was that home computers are in closer proximity to interference-prone equipment like TVs, short wave radio receivers and land mobile equipment. They also felt that class A devices are more carefully maintained and responsibly operated.

The two classes of computers face entirely different procedures to gain FCC approval. Class A devices need only verification. This is essentially an honor system in which the manufacturer tests his equipment to ensure that it meets FCC requirements, but doesn't need to notify the FCC. Certification, on the other hand, is an equipment authorization issued by the FCC for equipment designed to operate without individual license. The equipment receives a certification number which must appear on the computing device.

Manufacturers of class B devices have faced problems in two major areas: testing and modification. Both take time and money.

For example, while electromechanical compatibility (EMC) testing technology has been known and used for many years, an air of mystery still surrounds test procedures. Lately, only a few specialty labs have been doing EMC testing on the government's behalf.

Last June, in response to pressure

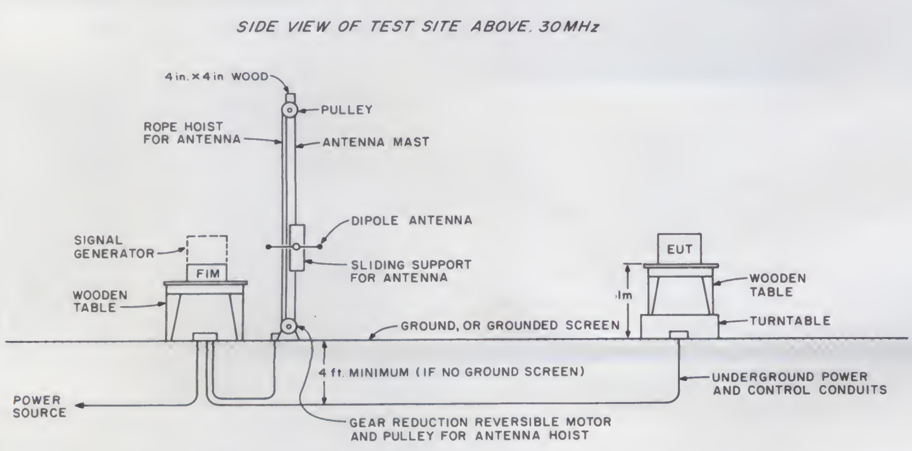


Fig. 1. *Typical test fixture used in radiated emissions testing. Site size can be three, ten or 30 meters.*

from the computer industry, the FCC published federal guidelines for EMC testing. It left little time for manufacturers to construct the elaborate test fixtures (see Fig. 1), buy the equipment and run the tests before the January 1, 1981, certification deadline.

The measurements themselves are simple. But the equipment is sophisticated and expensive. A spectrum analyzer capable of operating to 1 GHz costs $10,000–$30,000. Add to that the cost of building a test site complete with remote control turntable, calibrated dipole antennas and full ground screen, and smaller manufacturers find the project impossible.

Some manufacturers have turned to independent test labs, such as R and B Enterprises of Plymouth Meeting, PA. R and B specializes in all types of EMC testing and will perform the required tests (see Photo 1). Their fees vary, but a ballpark figure for class B testing is $1000 a session. Most micros can be tested in one or two sessions.

	Frequency (MHz)	Maximum RF Line Voltage (uV)
Class A devices:	0.45–1.6	1000*
	1.6–30	3000*
Class B devices:	0.45–30	250*
Class 1 TV devices:	0.45–25	100#

* 50 microhenry LISN
\# 5 microhenry LISN

Fig. 2a. RF voltage limits for power-line-conducted emissions.

Device	Frequency (MHz)	Distance (meters)	Field Strength (uV/m)
Class A:	30–88	30	30
	88–216	30	50
	216–1000	30	70
Class B:	30–88	3	100
	88–216	3	150
	216–1000	3	200
Class 1 TV:	30–47.7		15
	47.7–1000		15

Fig. 2b. RF voltage limits for radiated emissions.

Since radiation levels are affected by the physical placement of the computer, its disk drives, cabling, printer and so on, the tests must determine which configuration of hardware generates the most RFI. Also, micros that accept peripherals must be checked with each combination of equipment available. As far as the FCC is concerned, an 8K CPU that has the potential of being upgraded to 16K is actually two different computers. The 8K version may meet radiation requirements while the 16K doesn't. It is the manufacturer's re-

Industry Ordered To Clean Up Its Lines

Class B computing devices must meet two RFI requirements (see Figs. 2a and 2b). Power-line-conducted measurements determine the RF component from 450 kHz to 30 MHz present on the computer's ac power cord. Radiated VHF emissions determine the RF field radiated by the computer between 30 MHz and 1000 MHz. In both cases, a spectrum of frequencies is scanned within which any detected RF energy must be below specified values. Unfortunately, several variables which have a direct effect on the measurements must be accounted for if the results are to be accurate.

In power-line-conducted measurements, the impedance of the ac network the computer is connected to affects test results. Since the load placed on this network by other equipment is unpredictable, a line impedance stabilization network (LISN) must be connected in series with each supply conductor of the unit under test to provide standard test conditions. The LISN must be bypassed and filtered on the supply side so that adverse effects of supply impedance are minimized.

Some confusion exists as to what type of network should be used, since differing values of inductance affect RFI levels and frequencies. An LISN using a 50 uH series inductance with the line is now accepted as standard by the FCC (see Fig. 3).

Radiated VHF emissions can be measured in several ways but require the use of a qualified test site. To find out if his test site is qualified, a manufacturer must run tests to determine the site's attenuation. This is done by comparing signal attenuation levels with two identical, calibrated half-wave dipole antennas according to a specific, FCC approved procedure. Site size can be three, ten or 30 meters and both indoor and outdoor test sites are allowed, as long as they are constructed in a fashion the FCC finds appropriate (see Fig. 4).

Thirty-meter sites generally display the best attenuation characteristics but are the most costly to build. When tests are run indoors, a shielded room is suggested. (See Photo 1.)

—C. B.

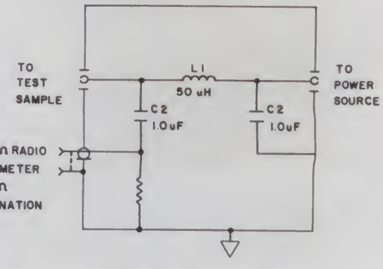


Fig. 3. LISN circuit used in power-line-conducted tests.

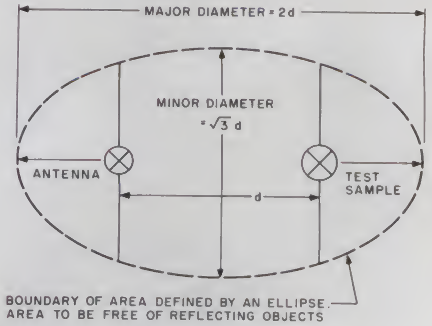


Fig. 4. Geometry of open field test site. Area within ellipse must be free of reflecting objects.

sponsibility to cover all bases.

The manufacturer must keep complete test records. If a test has to be repeated on demand, the hardware placement and the test equipment settings must be duplicated exactly. If a new peripheral is added to an already-certified system, the original system test must be run precisely as it was the first time.

At present, there is no government requirement for benchmark software to be used during the course of testing. The FCC realizes that an idling machine will have different interference characteristics than one that is performing I/O or arithmetic operations. As with system configuration, the burden of finding worst-case situations lies with the manufacturer. The price of oversight or deceit is the loss of certification and the right to market the product.

Modifications, too, can be expensive. In some instances, extra shielding will do the trick. But other manufacturers have had to redesign and modify the insides.

Combined, the testing and modifying has proven to be costly and time-consuming. In late fall of last year, several manufacturers filed for and received eleventh-hour deadline postponements, citing potential plant shutdowns, employee layoffs and severe economic hardship.

Unhappy Manufacturers

The FCC originally set the deadline for July 1, 1980. But the chore was simply too overwhelming. Many firms didn't have the equipment or the personnel, and weren't even too sure of what exactly the FCC wanted. As one engineer puts it, "Just interpreting the documents was a major effort." So responding to pressure from the industry, the FCC moved the class B deadline up to Jan. 1, 1981.

Even with the extra six months, many manufacturers found that they could not do all of the testing and make the necessary modifications in time. By late fall, the FCC found itself with a number of requests for postponement.

Ironically, one of the first was the Heath Company. The veteran of electronics manufacturing filed its letter on Oct. 20, stating that the task of meeting the new regulations was "monumental." The company claimed that it had spent nearly $375,000 on one product family alone.

Photo 2. A typical ten-meter outdoor test range. Note ground screen and adjustable antenna mast.

According to Heath technical consultant Steve Parker, each microcomputer has almost had to be redesigned. The units, he says, will look and operate identically, "but the insides will be entirely different."

"It's basically like developing a completely new product," he says. "Only we're not going to get anything out of it."

Parker says that Heath faces special problems because hobbyists make up such a large part of their market.

"Some companies can simply put a metal box around their computers," he says. "But ours have to be user-serviceable. Our customers need to get in there, especially if they're building a kit."

He cited other problems: "Apple, for example, doesn't have any video circuits. Our video circuits are creating more noise than anything else."

Parker won't say for certain that all of Heath's computers will be certified by April 1.

"Our engineers are working hard on it, and our legal department is working on getting extensions," he says. "We're optimistic that one way or another we'll be able to stay in production."

Apple, too, has had its share of expenses. One executive at Apple told

an industry newspaper in February that the company has spent $100,000 on testing equipment, added four people to the engineering staff and made "quite a sizeable change" in the Apple II.

People in the industry mention other problems. For instance, very little information on RFI test procedures is available. An *exo facto* ruling prohibits current FCC employees from discussing such matters with members of the industry, and the literature on RFI is scant. A two-day seminar in Boston sponsored by R and B drew over 100 computer hardware engineers, representing firms like Centronics, DEC, Itek and NEC. The company is running similar seminars in several other cities to dispense information on requirements and testing and on modification.

Also, companies are unable to make major purchasing commitments until their modified devices are approved. Thus, even though a firm might meet the April 1 deadline, it might be faced with a lag time between approval and production. This could severely disrupt production and marketing schedules.

Not everyone has experienced the problems Heath and Apple have. Radio Shack's line of TRS-80s received their certification by January, as did the Atari computers and Texas Instruments' TI 99/4. In fact, some technical people in the microcomputer industry feel that the obstacles have been vastly exaggerated.

"We designed our Color Computer from the start to be pretty much noise-free," says Steve Leininger, director of advanced development at Tandy/Radio Shack. "We, along with everyone else, knew that we were living on borrowed time."

Leininger adds that the expense of tests and modifications depends largely on the abilities of the technicians involved.

"To a digital computer designer, the solutions are not that obvious," he says. "To someone experienced in RFI, it's not that big of a deal."

B.J. Freeman, an engineer at Exidy Systems, agrees. An engineer with the proper background, he says, could do the job in six months; someone without the experience could take one and a half to three years.

"If the manufacturer gets someone with experience in magnetics, they shouldn't have any trouble modifying," Freeman says.

Leininger says that other computers, like the Atari, were originally designed to meet the requirements for a class 1 TV device. These standards are even stricter than those of class B.

Passing the Bucks

To the consumer, the impact of the new standards will be most noticeable when retail prices go up. Apple is adding $135 to its Apple II price. Heath's Steve Parker says Apple's figure "is in the ballpark."

In the long run, the consumer might be faced with a decrease in the number of new products being made. Some manufacturers will not find it profitable to continually release new models in the numbers and variety computer buffs are used to. Smaller manufacturers may simply not be able to allocate sufficient money and manpower to ensure that their equipment meets radiation requirements. This could eventually reduce the amount of research and development going on, since much of this activity is in the province of the small company.

In addition, some variety and flexibility of hardware may be lost as the number of products decreases and the possibility of equipment combinations becomes limited. For example, are Lobo drives approved for use with the Apple III or only the Apple II? Is an Exatron stringy floppy OK to use with a Radio Shack Model IV or Model V, or both?

Owners of two identical systems may not be able to swap parts, if they happen to be built before and after

Trying to Live in Harmony With Harmonics

By Frank J. Derfler, Jr.

There go the federal bureaucrats again. Getting into everybody's business and making the price of everything go up. Why did they come up with this Part 15 anyway?"

Contrary to what you might think, the FCC has used Part 15 for a long time to protect the frequencies of each radio service and to prevent the pollution of the radio spectrum. If you take a look at that old CB walkie-talkie in the bottom drawer, you will find it probably complied with Part 15. Garage door openers, radios and TV all fall under Part 15 of the Federal Communications Commission's rules governing low power communications and restricted radiation devices. Only in late 1979, after digital devices had been interfering with the radio services for several years, did they bring computing devices under their control.

The FCC issued their first report and order on computing devices in October 1979. In April 1980, as the result of many petitions, they extended their deadlines for compliance with emission restrictions until January 1981. They have divided microcomputers into two classes: those marketed for commercial and industrial use (class A) and those marketed for residential use (class B). The requirements are more strict for type B machines, because they are in closer proximity to television and radio receivers.

This is the same FCC that recently deregulated the telecommunications industry to foster business and competition. This is the same organization that cannot economically afford to police the citizen band radio spectrum for minor violators. The FCC is not a power-hungry empire-building bureaucracy. But the cross-hatch on the TV set can only be ignored for so long—the buzz on the police radio cannot be ignored at all.

The attention of the FCC was drawn to computer devices by complaints from police and fire departments,

Frank J. Derfler, Jr. (PO Box 691, Herndon, VA 22070) writes a monthly Dial-up Directory column for Microcomputing *on data communications.*

the new regulations. Thus, a businessman may own two identical Heath H-89s, but not be able to interchange boards.

We finally come to the issue of enforcement. Can the FCC put teeth into its certification stipulations in the real world?

The FCC has the authority to pursue cease-and-desist orders, or, in extreme cases, felony convictions. The latter could involve fines and prison terms of up to two years. Users, too, are liable, and can be ordered to shield their equipment. While it is unlikely that a cadre of RFI police will be raiding the homes of interference-generating computerists using unauthorized mixes of hardware, it is possible that manufacturers will be forced to design equipment in such a way that unapproved interconnects become difficult.

As fast as old problems are faced, new ones arise. As MITA points out, "changing work patterns will slowly blur the environmental distinctions between the home and office, and evolving communication services will continue to place additional demands on spectrum usage." What will the FCC do when more radio receivers are used in commercial environments? Or when more class A computers are used in residential areas?

Predictions can only be tentative. Recent commission actions come at a time when decontrol is a Washington watchword. How Republican President Ronald Reagan's new FCC chairman will view commission involvement in this elemental level of microcomputing remains to be seen.

But for the moment, no one in the industry is too pleased with the first chapter of the script. ■

The schedule of two-day seminars being conducted by R and B Enterprises to inform companies about the FCC-imposed computer specifications is as follows:
Baltimore—April 27/28
Boston—April 23/24
Chicago—April 27/28 and May 4/5
Denver—April 30/May 1 and May 6/7
Hartford—April 23/24

For more information contact R & B Enterprises, PO Box 328, Plymouth Meeting, PA 19462, (215)828-6236.

amateur radio operators and TV viewers. Coin-operated video machines were among the first to draw fire. Public watering holes that had spent considerable sums on high antennas and wide-screen TV projection systems found themselves with wide-screen wavy lines whenever someone slipped a coin in a video game.

I own a 1977-era microcomputer system. (A depreciated antique to everyone but the IRS.) When I'm on the system, my family cannot watch most of the TV channels because of radiated interference. This is not a new story. There are several very widely used commercial terminals which were radio frequency polluters long before microcomputers became available. Anyone who has ever tried to use a police scanner or amateur radio receiver near one of these systems knows how widespread the interference can be.

The RF radiation from microcomputers is particularly nasty because it is so varied, has so many harmonics and has such good antennas. Oscillators can be found on memory boards, CPUs, I/O ports and all through computer systems. They operate on many fundamental frequencies spread between 1 and 20 MHz. Little or no design effort was made to restrict the harmonics of their oscillation because computers run fast and fast speed means wide bandwidth. Strong signals in the spectrum above 300 MHz can be detected hundreds of feet away from many microcomputer systems. A typical microcomputer has a maze of wires and cables leading to peripheral devices. These leads make excellent antennas for RF radiation.

The leaky RF design of digital circuits can work both ways. It can let interference in as well as out. I have personally witnessed an electronic Christmas toy (a programmable tank) running under an operating table cause the complete crash of a TRS-80 Model I during a disk read. What goes out can come in.

Some portions of the industry were slow to respond to subtle warnings about the danger of inviting regulation by marketing devices producing so much interference. But other manufacturers, most notably Radio Shack and Atari, were able to produce new clean systems at reasonable prices. The Radio Shack TRS-80 Model III and Color Computer and the Atari 400 and 800 computers passed the certification tests run by the FCC. The Hewlett-Packard HP-85 and Texas Instruments 99/4 are also certified.

Both Heath and Apple have been granted waivers for their systems through April 1981. The folks at Heath say privately that they had to go for outside engineering help experienced in RFI design to clean up their line. Most of the changes were not in exterior packaging, but in design details. They probably found that little things such as cable routing, circuit board arrangement and ground plane shielding mean a lot. Word is that H-89 deliveries are being slowed by the changes.

The rumors persist that the Apple III has been delayed because of RFI problems. It was due for testing at the FCC laboratories in Maryland in February. Those of you with Apple II computers should know that a parts kit is available from Apple Computer to reduce the interference radiated from this system. The kit (part number 652-0152) has some capacitors for bypassing, ferrite beads for choking and solder lugs for grounding.

Bypassing, choking and grounding unwanted signals have been standard design practices of electrical engineers from the days of vacuum tubes. Several graduates of prestigious electrical engineering schools told me that today's students who elect to go into digital systems in their junior year are never exposed to those "RF techniques" at a practical level.

Shielding has also always been important in containing RF. Metal cabinets combining adequate shielding and cooling are expensive items. (Rumors also say Commodore is now benefiting from early problems with plastic cabinets which gave them a great metal cabinet line.) Some manufacturers are using a metallic compound called aqua-dag (usually found in TV picture tubes) to line the inside of their cabinets to reduce radiation.

In the end, the computing public will benefit from the FCC Part 15 enforcement. We will have equipment with new designs that will be both less interfering and less subject to interference from outside RF sources. Additionally, cleaner systems may bring peace to many residential computer users who now have to negotiate for air time. ■

This versatile machine-language program for the C1P saves any area of memory to the cassette recorder.

Save It with CASSY

By Andrew N. Messent

I had used my OSI C1P for a few months before I became aware of two major deficiencies in its ROM software.

First, it lacks a routine to save a machine-language program: the C1P ROM monitor can load a machine-language program but not save it. OSI markets an extended monitor that can save machine-language programs in a checksum format, but a saved program can only be loaded by the checksum loader in the extended monitor. The extended monitor also occupies over 2K of RAM, effectively leaving 1K for user programs (in a 4K system). Furthermore, there is no way to make a stand-alone back-up copy of the extended monitor.

Second, the BASIC SAVE command occasionally creates a tape that will not load correctly. Since the C1P has no cassette verification routine, the only way to check a tape is to load it; if the tape is bad the C1P gives a syntax error on any lines that wrap around to the next CRT line.

After spending many hours inputting and debugging a particularly long and complex program, I dumped out the data from a bad tape and discovered the C1P was inserting extra line-terminating characters at the point that the line wrapped around to the next CRT line. The line would load fine up to the extraneous characters, but the C1P then treated the second half as a new line and gave a syntax error because there was no line number.

I wanted to design a small machine-language program that could save any area of memory to the cassette recorder. It had to be small enough to be useful in a 4K system and easily input using the C1P ROM monitor. The tapes saved had to be auto-loading by the C1P ROM monitor's L command. There could be no software requirements other than the C1P ROM monitor. The program had to create tapes that transferred control back to the C1P ROM monitor after loading.

As designed, CASSY occupies 230 bytes of RAM at hex 0F00 to 0FE5 and uses seven page zero locations, hex 00B9 through 00BF. The program can easily be modified to transfer control to any memory location (even to the program that it is loading).

Implementation

The OSI C1P is a versatile machine with some nice surprises for an inquisitive programmer. The cassette interface is controlled by an MC6850 UART that is under software control.

The UART has two memory locations associated with it. The status and control registers are located at F000, and the read and write registers are at F001. A byte written to F000 goes into the control register; a byte read from F000 comes from the status register. A byte written to F001 goes into the transmit data register; a byte read from F001 comes from the receive data register. The UART status and receive data registers can't be written to, and, conversely, the control and transmit data registers can't be read.

To make use of the UART a certain sequence of steps must be followed.

1. Master-reset the UART.
2. Initialize the UART for the correct baud rate, number of data bits, number of stop bits, parity and interrupt logic.
3. Read the UART status and check the appropriate bit for the empty receive or transmit data register.
4. Read or write a byte of data.
5. Loop until all data has been transferred.
6. Flush the last two data bytes out of the UART.

Once you fully understand the UART operation and you also realize that the OSI ROM monitor's L command just transfers control to the cassette port, it is a simple matter to create a tape that controls the computer. That is essentially what an auto-load tape does.

The auto-load tapes are divided into three main parts: header, body and trailer. The header consists of the monitor commands to load the tape to the memory locations from which the recording was made. The body is the main part of the data, recorded a byte at a time followed by a carriage return. The trailer consists of the commands to return control to the ROM monitor.

Program Description

The program is broken down into a

Address correspondence to Andrew N. Messent, 1306 Pinewood Drive, Frederick, MD 21701.

main routine and three subroutines.

The subroutine that actually writes a byte to the cassette is called PUT. PUT reads the status register at F000 and loops until bit 2 is high, indicating that the transmit data register is empty. PUT then writes a byte to the transmit data register at F001 and returns to the calling routine.

PUT2 converts a byte of data into two ASCII characters and writes each, by calling PUT, to the cassette.

GETAD prompts for the starting and ending memory addresses to SAVE. GETAD uses two OSI ROM routines, FEED and LEGAL, at hex FEED and FE93, respectively. FEED accepts characters from the keyboard, and LEGAL checks to see if a valid hex character was entered. GETAD calls FEED and LEGAL and gets a pair of valid hex characters at a time. It converts each pair into a single hex byte, which is returned to the main routine.

CASSY calls GETAD, which prompts for the starting and ending addresses. The starting address is written to the cassette by PUT2 and PUT as an auto-load header, and is saved as the pointer to the current memory location. The memory contents to save are converted a byte at a time into two ASCII characters followed by a carriage return, which are then written to the cassette port.

The current memory location is then incremented and compared to the ending memory location. If they are not equal, the program loops back and processes the next byte. When all memory locations have been written, CASSY writes the commands onto the tape that returns control to the ROM monitor. CASSY then jumps to FE0C (the OSI ROM monitor's entry point).

Input CASSY Using the C1P's ROM Monitor

The easiest way to get CASSY up and running is to input it the way I did, a byte at a time, using the C1P ROM monitor. Power-up your C1P and hit the break key, followed by the M key. Then type in .0F00/. This procedure puts the monitor into the address mode, loads address 0F00 and then transfers the monitor to the data mode. Referring to the hex dump, each pair of characters is a byte. Starting at the upper-left corner of the dump and working to the right and down, type in a byte at a time, followed by a carriage return.

Go to the beginning and check the

Program listing. Assembly-language cassette save program for the OSI C1P.

```
;=========================================================================
;
;               * * *   C A S S Y   * * *
;
;                               WRITTEN 10-FEB-80 BY ANDREW N. MESSENT
;
;=========================================================================
;
;               A CASSETTE SAVE PROGRAM THAT CAN SAVE THE CONTENTS OF ANY
;       MEMORY LOCATIONS TO THE CASSETTE PORT. THE TAPE CREATED WILL BE
;       AUTO-LOADING AND WILL TRANSFER CONTROL TO THE ROM MONITOR UPON
;       COMPLETION.
;
        *=$0F00                         ;ASSEMBLE AT HEX 0F00
;
FEED    =$FEED                          ;C1P KEYBOARD INPUT ROUTINE
LEGAL   =$FE93                          ;C1P HEX VALIDATE ROUTINE
CASSY   LDY     #00
;
;               CURRENT CRT POINTER IS CONTAINED IN HEX 00BF AND 00BE
;
        LDA     #$45
        STA     $BE
        LDA     #$D3
        STA     $BF
;
;               GET THE START ADDRESS, STORE IN HEX 00BD AND 00BC
;
        JSR     GETAD
        STA     $BD
        JSR     GETAD
        STA     $BC
        LDA     #$2C
        STA     ($BE),Y
        INY
;
;               GET THE ENDING ADDRESS, STORE IN HEX 00BB AND 00BA
;
        JSR     GETAD
        STA     $BB
        JSR     GETAD
        STA     $BA
;
;               RESET THE UART
;
        LDA     #$03
        STA     $F000
;
;               INITIALIZE UART FOR CASSETTE OPERATION
;
        LDA     #$11
        STA     $F000
;
;               WRITE THE AUTO-LOAD HEADER TO THE CASSETTE
;
        LDA     #$2E                    ;AN ASCII '.'
        JSR     PUT
        LDA     $BD                     ;HI ORDER START ADDRESS BYTE
        JSR     PUT2
        LDA     $BC                     ;LO ORDER START ADDRESS BYTE
        JSR     PUT2
        LDA     #$2F                    ;AN ASCII '/'
        JSR     PUT
;
;               GET A BYTE OF MEMORY AT A TIME AND WRITE IT TO THE
;       CASSETTE  AS 2 ASCII CHARACTERS FOLLOWED BY A CARRIAGE RETURN
;
        LDY     #00
1$      LDA     ($BC),Y
;
        JSR     PUT2                    ;WRITE A BYTE
;
        LDA     #$0D                    ;WRITE A <CR>
        JSR     PUT
;
;               INCREMENT THE POINTER
;
        INC     $BC
        BNE     2$
        INC     $BD
2$      LDA     $BD
;
;               SEE IF DONE
;
        CMP     $BB
        BNE     1$
        LDA     $BC
        CMP     $BA
        BNE     1$
;
;               WRITE THE COMMANDS TO THE CASSETTE THAT WILL RETURN CONTROL
;       TO THE ROM MONITOR.
;               THE CODE HERE CAN BE MODIFIED TO TRANSFER CONTROL TO ANY
;       DESIRED ROUTINE, INCLUDING THE PROGRAM LOADING.
;
        LDA     #$2E                    ;AN ASCII '.'
        JSR     PUT
        LDA     #$46                    ;AN ASCII 'F'
        JSR     PUT
        LDA     #$45                    ;AN ASCII 'E'
        JSR     PUT
        LDA     #$30                    ;AN ASCII '0'
        JSR     PUT
```

More

Program listing continued.

```
        LDA     #$43                    ;AN ASCII 'C'
        JSR     PUT
        LDA     #$47                    ;AN ASCII 'G'
        JSR     PUT
;
;               FLUSH-THE LAST CHARACTERS OUT OF THE UART
;
        LDA     #00
        JSR     PUT
        JSR     PUT
;
;               RESET THE UART
;
        LDA     #$03
        STA     $F000
;
;               RETURN TO THE C1P ROM MONITOR
;
        JMP     $FE0C
;
;===========================================================================
;
;               * * *   E N D   O F   M A I N   * * *
;
;===========================================================================
;
;===========================================================================
;
;               * * *   S U B R O U T I N E S   * * *
;
;===========================================================================
;
;               * * *   G E T A D   * * *
;
GETAD   JSR     FEED                    ;GET A CHARACTER
        STA     ($BE),Y                 ;DISPLAY
        JSR     LEGAL                   ;VALID ?
        CMP     #$80
        BEQ     GETAD                   ;NO !
        INY                             ;CRT POINTER
        ASL     A
        ASL     A                       ;THE LO ORDER NIBBLE IS THE
        ASL     A                       ;HIGH ORDER NIBBLE OF THE
        ASL     A                       ;ADDRESS BYTE
        STA     $B9                     ;TEMP STORAGE
3$      JSR     FEED                    ;GET A CHARACTER
        STA     ($BE),Y                 ;DISPLAY
        JSR     LEGAL                   ;VALID ?
        CMP     #$80
        BEQ     3$                      ;NO !
        INY                             ;CRT POINTER
        AND     #$0F                    ;MASK OFF UPPER NIBBLE
        CLC
        CLD
        ADC     $B9                     ;ADD TO HI ORDER NIBBLE
        RTS
;
;===========================================================================
;
;               * * *   P U T   * * *
;
PUT     TAX
4$      LDA     $F000                   ;GET THE STATUS
        AND     #$02                    ;BIT 2 ONLY
        BEQ     4$                      ;NOT READY
        STX     $F001                   ;SEND A BYTE
        TXA
        RTS
;
;
;===========================================================================
;
;               * * *   P U T 2   * * *
;
PUT2    PHA
        AND     #$F0                    ;HI ORDER NIBBLE
        LSR     A
        LSR     A
        LSR     A
        LSR     A
        CMP     #$0A                    ;SEE IF ALPHA
        BMI     5$                      ;IF 0 - 9
        CLC
        CLD
        ADC     #$07                    ;IF A - F
5$      ADC     #$30                    ;TO ASCII
        JSR     PUT
        PLA
        AND     #$0F                    ;LO ORDER NIBBLE
        CMP     #$0A                    ;IF ALPHA
        BMI     6$                      ;IF 0 - 9
        CLC
        CLD
        ADC     #$07                    ;IF A - F
6$      ADC     #$30                    ;TO ASCII
        JSR     PUT
        RTS
;
;===========================================================================
;
;               * * *   E N D   O F   S U B R O U T I N E S   * * *
;
;===========================================================================
;
;
```

More

contents of all memory locations. CASSY should be ready to run a preliminary test. Type in .0F00G and 0300. The program will give a right arrow. Type in 0310. In about ten seconds the screen should clear and memory location 0000 will be displayed in the upper-left corner of the CRT. If that ran correctly, the ROM monitor will be back in control. If CASSY did not operate as described, go back and verify all memory locations. Make sure that you didn't transpose any digits, type in a B for an 8 or make a similar error.

CASSY should be ready to put to work. The first program to save is CASSY. Set up the cassette recorder to record, with a blank tape advanced past the leader. Type in .0F00G followed by 0F00. Start the recorder and type in 0FFF. In about one minute when the screen clears and memory location 0000 is displayed, shut off the recorder. You have just made an auto-load copy of CASSY.

You must try to load the tape to check it. Set up the cassette recorder to playback, hit the break key and type in M. Start the cassette recorder playing back and type in L as soon as the tape has advanced past the leader. The memory locations and contents will be sequentially displayed in the upper-left corner of the CRT. The program is loaded when control has been returned to the ROM monitor.

CASSY can be used to SAVE any memory locations to the cassette port. If you are going to use CASSY with BASIC programs, you must also record the contents of memory locations 007B through 0080, which point past the end of the program to the available working storage that BASIC uses. When you load the BASIC program, you must restore those values to their original locations; otherwise, when you run the program, BASIC will start over-writing it with the program variables. You will have to limit the memory size if you will be running BASIC at the same time CASSY is in memory.

Summary

The listing and hex dump presented here locate CASSY at 0F00 to 0FFF. I am currently maintaining two copies of CASSY, one located at 0300 and the other at 0F00. CASSY can be relocated by changing the addresses after each JSR command. Make sure you don't change the addresses of FEED and LEGAL, since they are OSI's

Program listing continued.

```
;=========================================================================
;
;             * * *   H E X   D U M P   O F   C A S S Y   * * *
;
;       HEX DUMP OF CASSY LOCATED AT 0F00
;
;       ==========================================================
;       START   :                 MEMORY CONTENTS
;       ==========================================================
;       0F00    ! A0 00 A9 45 85 BE A9 D3 85 BF 20 8D 0F 85 BD 20
;       0F10    ! 8D 0F 85 BC A9 3E 91 BE C8 20 8D 0F 85 BB 20 BD
;       0F20    ! 0F 85 BA A9 03 8D 00 F0 A9 11 8D 00 F0 A9 2E 20
;       0F30    ! B4 0F A5 BD 20 C1 0F A5 BC 20 C1 0F A9 2F 20 B4
;       0F40    ! 0F A0 00 B1 BC 20 C1 0F A9 0D 20 B4 0F E6 BC D0
;       0F50    ! 02 E6 BD A5 BD C5 BB D0 EA A5 BC C5 BA D0 E4 A9
;       0F60    ! 2E 20 B4 0F A9 46 20 B4 0F A9 45 20 B4 0F A9 30
;       0F70    ! 20 B4 0F A9 43 20 B4 0F A9 47 20 B4 0F A9 00 20
;       0F80    ! B4 0F 20 B4 0F A9 03 8D 00 F0 4C 0C FE 20 ED FE
;       0F90    ! 91 BE 20 93 FE C9 80 F0 F4 C8 0A 0A 0A 0A 85 B9
;       0FA0    ! 20 ED FE 91 BE 20 93 FE C9 80 F0 F4 C8 29 0F 18
;       0FB0    ! D8 65 B9 60 AA AD 00 F0 29 02 F0 F9 8E 01 F0 8A
;       0FC0    ! 60 48 29 F0 4A 4A 4A 4A C9 0A 30 04 18 D8 69 07
;       0FD0    ! 69 30 20 B4 0F 68 29 0F C9 0A 30 04 18 D8 69 07
;       0FE0    ! 69 30 20 B4 0F 60 00 00 00 00 00 00 00 00 00 00
;       0FF0    ! 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
;       ==========================================================
```

ROM routines.

An enterprising individual could supply any number of enhancements to CASSY. How about named files? CASSY could be modified to prompt for a file name and write some sort of header to the cassette before the actual file. You would have to create a LOAD program to load the correct file, but it would not be difficult. To load a file, the program would search the tape for the correct file header before transferring control to the ROM monitor. The ROM monitor would then just load as normal. Now if someone could come up with a cassette deck control port that could operate the cassette deck motor. . . . ■

Call For Manuscripts

Kilobaud Microcomputing is looking for business articles!

Businessmen in all fields are beginning to take notice of the microcomputer. They are eager to know which computers, peripheral equipment and applications software will let them take full advantage of this new tool. What knowledge do you have to share?

Here are the kinds of articles that we want you to write for us:

- Are you a businessman with a system up and running? We want to know how it works. What were your expectations? Have they been fulfilled? Did you find the software that you wanted? What problems have you had? How did you overcome them? What recommendations do you have for other businessmen?
- We want reviews from a businessman's perspective of specific hardware and software. If you've recently bought a new product and want to tell others how great—or poor—it is, *Microcomputing* will provide you with a forum.
- What programs have you written to meet your specific needs? Perhaps another businessman can use them, too. Even if he can't, your program may serve as a springboard for other ideas.
- Perhaps you aren't using your micro for business, but know a company that is. Trot on down with your pencil and notebook, and find out what they're up to. While they might not have the time to write up their experiences, they might be more than willing to tell somebody else about them. And an outside observer will often be able to see things with a unique and valuable perspective.

Don't worry if you're not a professional writer. That's what we editors are here for. And we'll be more than happy to send you a copy of our writer's guidelines.

Send your manuscripts and correspondence to:
Kilobaud Microcomputing
Pine St.
Peterborough, NH 03458

Quicksilver Micro System

By Peter W. Marcus

A microcomputer lends itself well to monitoring the environment. With this in mind I decided to use my microcomputer evaluation kit to record and display the present temperature and the low and high extremes since last reset.

Design Considerations

Such a system requires a temperature transducer and some form of analog-to-digital converter to interface to the computer. Initially, I experimented with thermistor temperature transducers, but these have a nonlinear temperature-versus-voltage response and thus are difficult to use over wide temperature ranges. Therefore, I decided to use the LM334 (Photo 1).

This is a three-terminal integrated circuit, the output voltage of which varies linearly with temperatures from 32 to 158 degrees Fahrenheit. It is connected as a current source; thus, its output is independent of supply voltage variations, and the resistance of long connecting wires does not affect transducer output.

A voltage-to-frequency converter, along with software, does the analog-to-digital conversion. An advantage of this method is that it requires only one bit of input port. Also, the analog-to-digital converter board (Photo 2) can be conveniently located away from the computer, since only two wires are needed to carry the serial data.

System Operation

The system operates as follows (see Fig. 1): The LM334 (IC1) outputs a current proportional to the temperature. This current is converted into a voltage proportional to the temperature, across the load resistor (R1). This voltage is buffered by op amp IC2a and offset as required for calibration by op amp IC2b.

The voltage, proportional to the temperature, is fed into the input (pin 7) of the LM331 voltage to the frequency converter (IC3). The LM331 puts out narrow negative trigger pulses at a frequency proportional to temperature. These pulses are too narrow for detection by the computer, so a D-type flip-flop (IC4) is connected in a divide-by-two mode at the output of the voltage-to-frequency converter. A 50 percent duty cycle wave, which gives adequate pulse widths, is obtained.

Using both the offset adjustment (R2) and slope adjustment (R3), a square wave whose frequency equals degrees Fahrenheit times ten is obtained at the output of the D-type flip-flop (IC4). This square wave is fed into an input port of the computer for processing.

Note that the zero offset voltage is derived from pin 2 of the LM331 (IC3), which supplies a highly stable 1.9 V reference. This makes the zero offset independent of the temperature and power supply variations.

Other factors making the circuit accurate and temperature independent include low temperature coefficient resistors (1 percent resistors) and capacitor (polypropylene capacitor C1), 15 turn cermet potentiometers (R2

Photo 1. The LM334 temperature transducer, with resistor and twisted pair leads soldered to it for remote operation. (Photos by Jean Fazio)

Peter W. Marcus, 10225 Coral Way, Apartment B-122, Miami, FL 33165.

and R3) for precise adjustability, and the use of the highly linear and temperature-stable LM331 voltage-to-frequency converter.

I got the design information on the linear integrated circuits from the *1978 National Semiconductor Linear Databook.*

Software

It was necessary for the computer to count frequency divided by ten rather than direct frequency. The initial output from the temperature-to-frequency conversion circuit can be a zero or one, which causes a ±1 count fluctuation when frequency is measured directly. Therefore, the circuit was designed to output a square wave at a frequency equal to ten times the ambient temperature.

For example, at 70 degrees F the frequency is 700 Hz. Thus, the computer simply measures the frequency divided by ten, with the result equal to degrees Fahrenheit. The computer then compares the temperature with the previous minimum and maximum value and stores any new minimum or maximum. The computer used was the Motorola MEK6800D2 kit (Photo 3).

The program is set up so that the computer spends one second measuring temperature (frequency/10) and three seconds in which the low, current and high temperatures are output sequentially. To measure frequency divided by 10, the program polls the input port for a change in level. After 20 changes in input level (10 cycles), a count is stored. This process is continued for one second, making the result the frequency in cycles-per-second divided by 10. The one second window is set by going through the polling loop a fixed number of times, taking up a total of one second. Dummy timing steps are used so that the polling loop takes the same number of machine cycles independent of whether no level change, a level change or a count is recorded by the computer (Fig. 2).

Another important consideration in measuring frequency with software polling is that the polling loop must repeat fast enough so that level changes are not missed. The flowchart in Fig. 2 includes temperature (frequency divided by a scaler) measurement and output of minimum, present and maximum temperature since last reset. This scheme can be applied to any computer that will run machine code routines, and has at least one bit of input port and a means of displaying results.

Listing 1 gives the detailed Motorola 6800 listing for the MEK6800D2 kit. The software timing and counting loops are based on the system's .6144 MHz clock frequency. The system 6821 peripheral interface adapter is buffered and drives the displays (seven segment LEDs) on the D2 kit. The program uses two of these displays to output temperature readings. Bit 7 (PA7) of the user 6821 peripheral interface adapter is used to input the frequency signal, since bit 7 is easily tested in 6800 systems.

The program is position independent, meaning it can be relocated in memory without any changes. Thus, permanent storage can be accomplished by simply loading the pro-

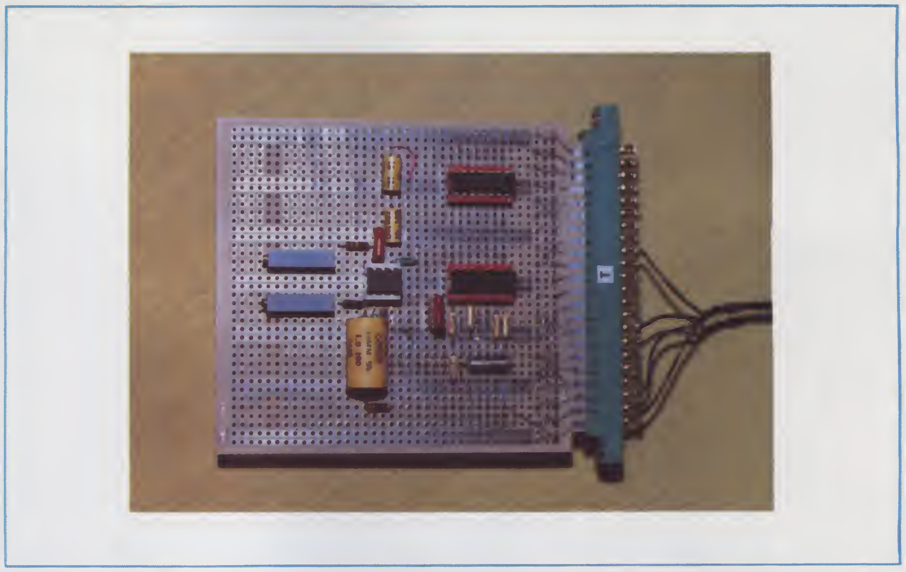
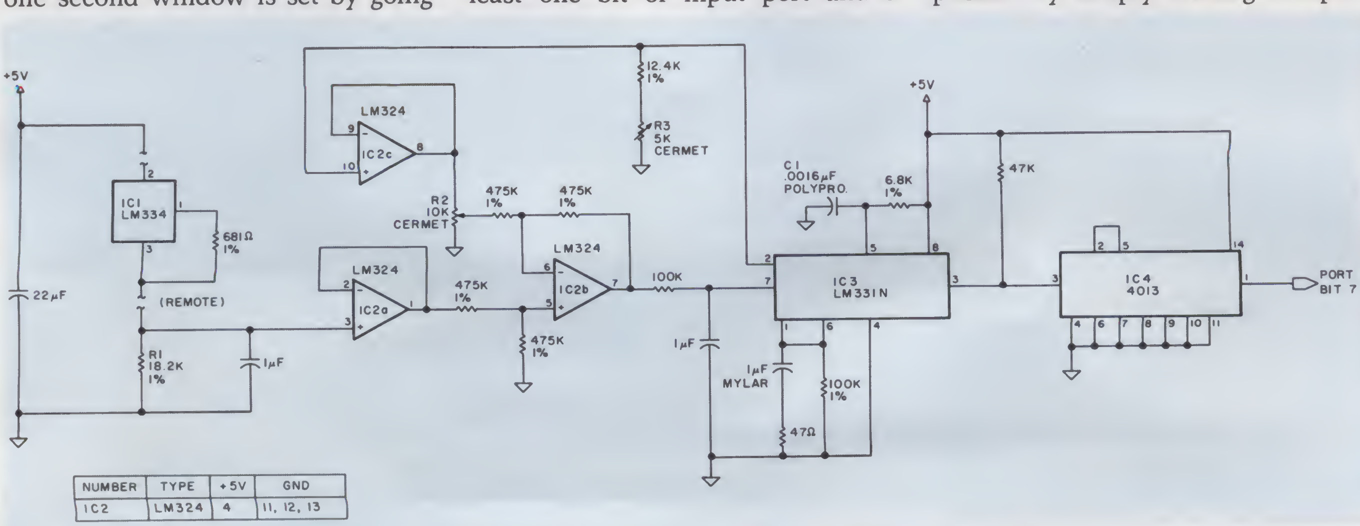


Photo 2. Temperature-to-frequency conversion circuit using the LM331 voltage-to-frequency converter and low temperature coefficient components. The LM334 temperature transducer (not shown above) is located remotely from the board.



NUMBER	TYPE	+5V	GND
IC2	LM324	4	11, 12, 13

Fig. 1. A circuit which outputs a frequency equal to degrees Fahrenheit times ten.

gram (Listing 1) into EPROM and using one of the extra EPROM sockets on the D2 kit.

Final Comments

By insulating the transducer with plastic tubing and immersing it in water of varying temperature, I found that the temperature readings were well within the accuracy of a common household thermometer for the full 32 degrees F to 158 degrees F operating range of the transducer. Extended range versions of the transducer (LM134, LM234) are available if larger temperature ranges are re-

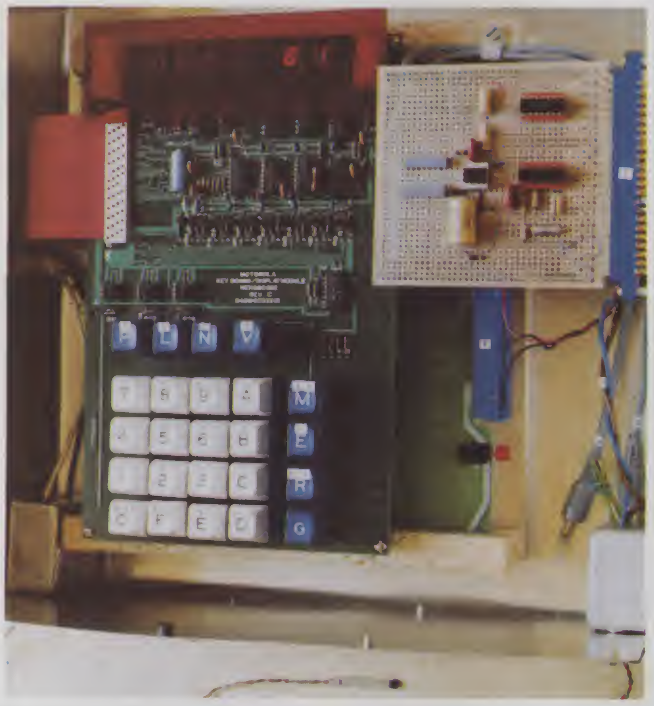


Photo 3. The complete temperature monitoring system. The MEK6800D2 computer kit and temperature to frequency conversion circuit are in an aluminum box (cover removed) with radio frequency suppressors leading to external power supply and temperature transducer connections to eliminate interference with television stations. The temperature transducer is in the foreground.

Listing 1. The 6800 assembly listing for a thermometer which displays the present temperature as well as the minimum and maximum temperatures since last reset.

```
                     SYMBOL TABLE

                     TMIN     =$0000       Minimum temperature
                     TMAX     =$0001       Maximum temperature
                     IPRES    =$0002       Present input level
                     IPREV    =$0003       Previous input level
                     COUNT    =$0004       Count & present temperature
                     SCAL     =$0005       Scaler
                     TBUFF    =$0006       Temporary storage for temperature
                     DIGI     =$0007       Digit 1 buffer
                     DIG2     =$0008       Digit 2 buffer
                     PORT     =$8004       I/O port A & data direction register
                     CR       =$8005       Control register A

                     CONFIGURE I/O PORT A AS AN INPUT

0100  B6 80 05              LDAA CR        Load control register
0103  84 FB                 ANDA #$FB
0105  B7 80 05              STAA CR        Access data direction register
0108  86 00                 LDAA #00
010A  B7 80 04              STAA PORT
010D  B6 80 05              LDAA CR
0110  8A 04                 ORAA#$04
0112  B7 80 05              STAA CR        Access output register

                     MEASURE TEMPERATURE (FREQUENCY/10)

0115  86 99                 LDAA #$99
0117  97 00                 STAATMIN       Initialize low temperature
0119  7F 00 01              CLR TMAX       Initialize high temperature
011C  CE 27 58   TMESET     LDX #$2758     Initialize timer
011F  7F 00 04              CLR COUNT      Reset counter
0122  86 14                 LDAA #$14
0124  97 05                 STAA SCAL      Initialize scaler (10 cycles/count)
0126  B6 80 04   INPUT      LDAA PORT      Read input port level
0129  97 02                 STAA IPRES     Store present input
012B  98 03                 EOR IPREV      Poll for input change
012D  2A 12                 BPL DELAY1     Branch if no input change
012F  7A 00 05              DEC SCAL       Decrement scaler
0132  26 11                 BNE DELAY2     Branch if scaler not zero
0134  86 14                 LDAA #$14
0136  97 05                 STAA SCAL      Set scaler
0138  96 04                 LDAA COUNT
```

More →

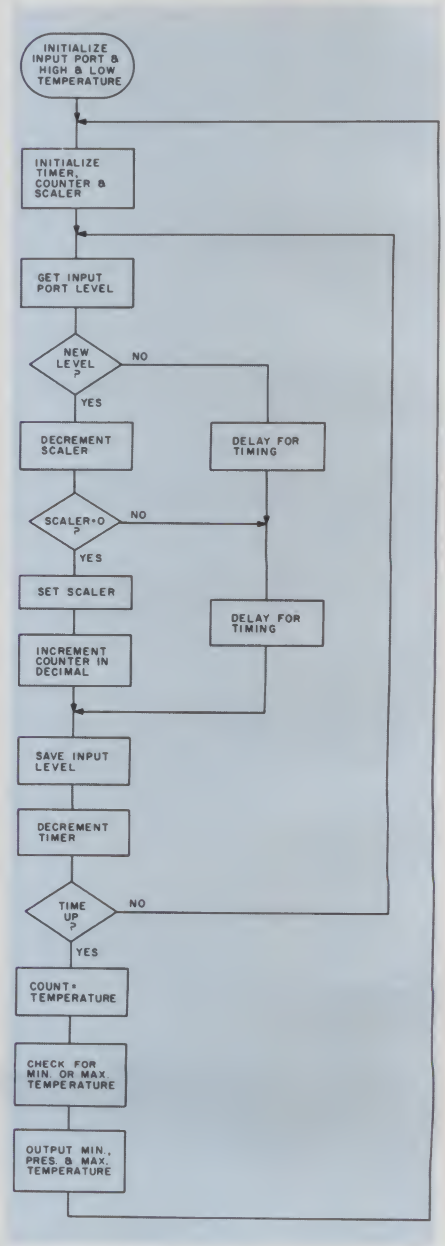


Fig. 2. Flowchart of the machine-language program that counts a frequency which is equivalent to temperature. The present temperature and maximum and minimum temperature since last reset are displayed.

Listing 1 continued.

```
013A  8B 01              ADDA #$01
013C  19                 DAA            Increment count in decimal
013D  97 04              STAA COUNT     Store count
013F  20 0A              BRA CONT       Branch to CONT
0141  A1 00     DELAY1   CMPA 0,X
0143  A1 00              CMPA 0,X       Dummy steps for timing
0145  6D 00     DELAY2   TST 0,X
0147  6D 00              TST 0,X
0149  6D 00              TST 0,X        Dummy steps for timing
014B  96 02     CONT     LDAA IPRES
014D  97 03              STAA IPREV     Store input
014F  09                 DEX            Decrement timer
0150  26 D4              BNE INPUT      Recycle if time not up

                CHECK FOR MINIMUM OR MAXIMUM TEMPERATURE
                AND DISPLAY RESULTS

0152  96 04              LDAA COUNT     Load count (temperature)
0154  91 01              CMPA TMAX      Compare temperature to previous max.
0156  25 02              BCS CHKMIN     Branch if not max. temperature
0158  97 01              STAA TMAX      Store new maximum temperature
015A  91 00     CHKMIN   CMPA TMIN      Compare temperature to previous min.
015C  24 02              BCC OUTPUT     Branch if not min. temperature
015E  97 00              STAA TMIN      Store new minimum temperature
0160  96 00     OUTPUT   LDAA TMIN      Load minimum temperature
0162  8D 0D              BSR DISPLY     Branch to display subroutine
0164  96 04              LDAA COUNT     Load present temperature
0166  8D 09              BSR DISPLY     Branch to display subroutine
0168  96 01              LDAA TMAX      Load maximum temperature
016A  8D 05              BSR DISPLY     Branch to display subroutine
016C  20 AE              BRA TMESET     Go to next measurement
016E  01                 NOP
016F  01                 NOP
0170  01                 NOP

                DISPLAY SUBROUTINE (CONVERTS BINARY CODED
                DECIMAL TO SEVEN SEGMENT OUTPUT)

0171  97 06     DISPLY   STAA TBUFF     Store temperature
0173  84 0F              ANDA #$0F      Mask upper 4 bits
0175  4C                 INCA
0176  CE E3 C9           LDX #$E3C9     Load X with digit table address - 1
0179  08        INCX1    INX
017A  4A                 DECA           Set pointer to 7 segment digit
017B  26 FC              BNE INCX1
017D  A6 00              LDAA 0,X       Load 7 segment digit
017F  97 07              STAA DIG1      Store temperature (low digit)
0181  86 04              LDAA #$04
0183  74 00 06  SHFTBF   LSR TBUFF
0186  4A                 DECA
0187  26 FA              BNE SHFTBF     Get temperature (high digit)
0189  96 06              LDAA TBUFF     Load temperature (high digit)
018B  4C                 INCA
018C  CE E3 C9           LDX #$E3C9     Load X with digit table address - 1
018F  08        INCX2    INX
0190  4A                 DECA           Set pointer to 7 segment digit
0191  26 FC              BNE INCX2
0193  A6 00              LDAA 0,X       Load 7 segment digit
0195  97 08              STAA DIG2      Store temperature (high digit)
0197  CE 2A 00           LDX #$2A00     Load output delay
019A  96 07     DISOUT   LDAA DIG1      Load segments (low digit)
019C  B7 80 20           STAA $8020     Segments into display port
019F  86 01              LDAA #$01
01A1  B7 80 22           STAA $8022     Enable digit 1 cathode (port)
01A4  20 03              BRA $03        Delay for brightness
01A6  01                 NOP
01A7  01                 NOP
01A8  01                 NOP            Optional delay for brightness
01A9  7F 80 22           CLR $8022      Disable digit 1 (cathode)
01AC  96 08              LDAA DIG2      Load segments (high digit)
01AE  B7 80 20           STAA $8020     Segments into display port
01B1  86 02              LDAA #$02
01B3  B7 80 22           STAA $8022     Enable digit 2 (cathode)
01B6  20 03              BRA $03        Delay for brightness
01B8  01                 NOP
01B9  01                 NOP
01BA  01                 NOP            Optional delay for brightness
01BB  7F 80 22           CLR $8022      Disable digit 2
01BE  09                 DEX            Decrement timer
01BF  26 D9              BNE DISOUT     Branch if time not up
01C1  39                 RTS            Return to main program

                7 SEGMENT DIGIT TABLE IN JBUG MONITOR

E3CA  40        DIGTBL   $40            0
E3CB  79                 $79            1
E3CC  24                 $24            2
E3CD  30                 $30            3
E3CE  19                 $19            4
E3CF  12                 $12            5
E3D0  02                 $02            6
E3D1  78                 $78            7
E3D2  00                 $00            8
E3D3  18                 $18            9
```

quired. For applications where rapid temperature measurement is needed, a method using a standard eight-bit analog-to-digital converter would be required.

The temperature measurement method described in this article could be used in more complex applications such as temperature controllers or home energy management.

For a stand-alone system, the MEK6800D2 kit could be replaced by a small 6802, 6821 board such as the one described by Peter Stark in the June 1980 issue of *Microcomputing* ("Kilobaud Klassroom," p. 22). Since both systems use the 6821 peripheral interface adapter for input-output and run 6800 code, this would be very straightforward. The addition of two seven-segment displays and drivers to the 6802, 6821 board and minor software modifications is all that would be required.

The LM331 voltage-to-frequency converter is available from Priority One Electronics and Jameco Electronics. The LM334 temperature transducer-current source can be purchased from Radio Shack, Jameco or Digi-Key.■

The author makes a strong case for Pascal over BASIC.

Pascal Means Business

By Mark J. Borgerson

Pascal should be the language of choice for anyone writing or using microcomputer programs in business—and I'm going to prove it to you.

Why Pascal? For the sample programs listed here, I couldn't have done it any other way.

First, Pascal gives me more memory space and a more compact code to run in that space—allowing larger data arrays to be kept in memory.

Second, it gives me more data space per diskette—140K per diskette vs 110K in BASIC with Apple DOS 3.2. It also permits much more efficient use of the storage space.

And third, Pascal allows faster execution, important in such tasks as sorting client names alphabetically.

These qualities benefit both the programmer and the user. Other advantages, primarily for the programmer, are that:

- Structured programming is supported, even required (unless you change compiler options).
- Data structures can be defined by the programmer.
- A good editor and file-handling system is available to make writing and altering source code more convenient than is the case with Applesoft BASIC.

The sample programs will help you if you are a programmer. If you are a businessman, the main points of interest will be the descriptions of the operator interface, the storage capacity and the cost.

I have a suggestion for the latter group. Read the article, and then find a competent programmer. Show him the sample programs and ask whether they make any sense without the explanatory text. If the programmer asks, "Uh, what language is this written in?", start looking for another programmer. If the programmer is a relative, spouse or friend, buy them a good book on UCSD Pascal. There's a list of recommended reading at the end of the article.

CFILE

The CFILE programs are designed to help a representative of a brokerage house keep track of his clients and their transactions. In essence, they replace an address book and a number of transaction record books.

Like any other set of data-base management programs, the CFILE system has three main parts:

- Procedures to let the user enter information into the system.
- Procedures for processing and storing the entered information.
- Procedures for recalling information desired by the operator and presenting it either on the visual display or the line printer.

Unlike a general-purpose data-base management program, the CFILE programs include only those types of data that will be used by the broker's representative. The programs can't be used with completely different types of data or functions. The customer gets a program that does what he or she requires without paying (in increased program size or reduced efficiency) for the ability to balance Sally's checkbook or keep track of Jim's recipe file.

Of course, in many personal and some business applications, there is something to be said for the ability to sell the same program several hundred times, thus cutting the cost per copy. One of the major advantages of business programs written in Pascal is that you will soon be able to find a data-base management program which very nearly fits *your requirements*, and then hire a programmer to make the changes needed to make the result *exactly* what you need.

Which brings up a point: If you can afford it, hire a programmer to make the computer do things your way! If you can spend $5–15,000 for hardware, you can pay a programmer $2–4000 to make sure that the hardware does what you want it to do.

The inventory management requirements of a TV and stereo shop are quite different from those of a jewelry store. No one program can serve both interests with maximum

Address correspondence to Mark J. Borgerson, 2015 NW Circle Blvd., Corvallis, OR 97330.

Main Program	Menu Choices	Files Used	
ENTRY	Add New Client	NAMEFIL	
	Enter Client Transactions	PERSFIL	
	List Clients by Name	TRANSFIL	
	Edit Client Data	SNAMEFIL	
	End Program		
REVIEW	Review Client Transactions	NAMEFIL	
	List Clients by Name	PERSFIL	
	Review Client Portfolio	TRANSFIL	
	End Program	SNAMEFIL	
UTILITIES	Backup Transactions Disk		
	Backup Clients Disk	NAMEFIL	BNAMEFIL
	Print Client Directory	PERSFIL	BPERSFIL
	List Stock Names in Use	TRANSFIL	BTRANSFIL
	List Holders of a Stock	SNAMEFIL	BSNAMEFIL
	Print Mailing Labels		
	Set and Review Client Lists		

Fig. 1. Block Diagram of the CFILE Programs.

efficiency. Automating any record-keeping procedure is going to be a lot of work. Why make things more difficult by having to learn someone else's terminology and operating *methods?*

The CFILE program system has three major segments, shown in Fig. 1 with the program operations they support. Each is run as a separate program, but all three use the same data base and all agree on the types of data in the files.

The data is stored on two floppy diskettes. One contains the information on the clients, such as names, addresses and telephone numbers. The other stores the transactions for the clients and a list of the stock names used in the transactions.

Since the customer has only two disk drives, both of which are used for data while a program is running, the programs will prompt him to put the program diskette back into one of the drives when switching from one module to another. The programs will now allow the following amounts of data to be stored on a single set of diskettes:

Maximum number of clients:	750
Maximum total transactions (for all clients):	4500
Total number of different stock names:	1000

After six months of use, the diskettes contain about 400 clients, 1800 transactions and 390 different stock names on the data diskettes.

A frequent argument against Pascal is that its input/output abilities are insufficient for many tasks. Before the UCSD version let the language use variable-length character strings and read and write random-access files, I might have agreed. But the I/O capabilities of Apple Pascal are now every bit as good as those of Applesoft BASIC. In fact, I know one Apple user who has an external terminal, a line printer, dual disk drives, a modem and a digital plotter. He happily uses all these peripheral devices with Pascal—in fact, he reports that he seldom uses BASIC at all. He says it's too difficult keeping track of all the peripherals in BASIC.

It is true that Apple Pascal places some constraints on which peripherals may be placed in which slots in the motherboard. But this seldom turns out to be a problem in business applications.

In terms of the type of input/output most immediately visible to the business user—keyboard input and video output—Pascal is at least equal to BASIC. The Pascal operating system lets the programmer collect data from the keyboard, check it for errors and put the information on the video screen with all the flexibility allowed by BASIC.

Inputting a number in Pascal with strict checking for legal characters (0 to 9, +, – and .) does require a fair amount of source code. However, that code need be written only once. The key to the error-checking in this code is the ability to define the set of characters which are legal in the entry of a number. Compare the BASIC and Pascal listings in Example 1.

Both examples have adequate error-checking and both can be understood from the comments in the source code. With examples of this length, there is little difference in clarity. There is one big advantage to the Pascal code—it can be reused in any program without change. You simply use your text editor to insert the code into your source file. To use the BASIC version, you would probably have to change the line numbers and make sure that no other part of the program uses the variable A$—or if it is used, that it is OK to mess it up a little.

I have only a little to say about formatting output to the video display. From the programmer's point of view, output formats will always be a large part of his work. Apple Pascal makes some things easier and some things more difficult than Applesoft. For instance, Pascal has no intrinsic tab function. It is possible to write a general-purpose tab function, but I found it easier to simply build up an output line piece by piece. Pascal allows better control of integer and real number output formats than does Applesoft. For instance, the statement

```
Writeln('A = ',A:8:2);
```

will write the value of A in a field eight spaces wide with two digits after the decimal point. The Applesoft code to accomplish the same result is considerably more complex. A very important feature of Apple Pascal when writing business programs is the ability to store integers of impressive size. It is possible to define an integer variable with as many as 36 digits. In business programs an integer length of 11 digits lets you use numbers up to 999 million dollars without losing any pennies in roundoff errors. Applesoft programmers start losing their cents at this point.

The time has come to discuss the greatest strength of Pascal—the ability to create special types of variables to fit specific program requirements. I'll start with a simple example: a calendar date in the form mm/dd/yy. We see dates in this form every day, but we don't always consider the possible ways in which a computer might store this information. There are a number of possible alternative storage formats. First, we might simply store the date as a string. This would generally require eight bytes (maximum) for the characters and one byte to define the length of the string. Another alternative (in both BASIC and Pascal) is to store the month, day and year as separate integers. This would reduce the storage needed to six bytes, two for each integer (I am ignoring for the moment any variable

BASIC

```
 10 REM GET KEYBOARD ENTRY AND CHECK TO SEE IF
 20 REM IT IS LEGAL FOR NUMERIC ENTRY.
 30 GET A$
 40 IF ASC(A$)>ASC("0") AND ASC(A$)<ASC("9") THEN GOTO 100
 50 IF A$="+" OR A$="-" OR A$="." THEN GOTO 100
 60 REM YOU GET HERE ONLY IF ERROR
 70 REM BEEP SPEAKER AND RETURN NULL STRING
 80  A$=CHR$(0):PRINT CHR$(7);:RETURN
100 REM LEGAL CHARACTER
110 RETURN
```

Pascal

```
function GETNUMCHAR:char;
(* returns a valid numeric character *)
(* or a null if invalid entry *)
var numset:set of char;
      achar:char;
begin
   numset:={'0'..'9','+','-','.'};
   read(keyboard, achar);
   if achar in numset then
      getnumchar:=achar;
   else
      begin
       write(chr(7)); (*beep speaker*)
       getnumchar:=chr(0); (*return a null*)
   end;
end; (*getnumchar*)
```

Example 1.

name storage and pointers). If we were really desperate for more efficient storage, we might use some packing technique, such as MD%=1000*MM+DD, to pack both the month and day into a two-byte integer. If we do much manipulation of the date information, the computational overhead may become significant.

Pascal allows us a much more convenient way to manage date information. We can simply define a logical record to contain exactly the type of information present in a date. Here is definition of a date record:

```
TYPE
  Month=1..12;
  Day=1..31;
  Year=0..99;

DATREC=packed record
  mm:Month;
  dd:Day;
  yy:Year;
end; (*DATREC*)
```

In the first part of this variable type declaration we set the allowable limits for the values for each part of a date record. This has two significant effects. First, the program will come to a screeching halt should any part of the date record be outside the defined limits. This forces the programmer to carefully check each part of the date to see that input is within limits. It also guarantees that no nonsense data will end up in the files.

The second effect of defining the numeric limits is that when the components are made part of a packed record, the compiler will reserve only as much space as is needed to store a number of the defined size. You can see how that affects the date record in Example 2.

As you can see, it is possible to pack a complete date into a single 16-bit data word (Pascal uses the 16-bit word as the smallest possible element of data storage). Furthermore, the Pascal operating system will take care of all the packing and unpacking operations. Oh, I don't mean to say that the computational overhead disappears—it doesn't. It's simply that it is no longer the programmer's concern—the system does it all, and much faster than a BASIC program could. Listing 1 shows the Type and Variable declarations from one of the major sections of the CFILE programs. As you can see, a number of specially defined variable types are used in these programs. I will also discuss the file structures and arrays shown in this listing a little further on in this article.

Specially defined variable types also lead to another convenience when it comes time to take the data out of storage and manipulate it. Suppose we wish to set the value of a date DATE1 to today's date (which we have already entered from the keyboard). In Pascal it would look like this:

```
(*assume DATE1 has already been defined as a DATREC*)
DATE1:=TODAY;
```

All the elements of the date record are transferred by simply referencing the single variable name. Suppose that we were using BASIC to store the date as an array of three integers. The same transfer would look like this:

```
10 FOR I=1 TO 3
20 DT%(I)=TD%(I)
30 NEXT I
```

Not quite so convenient, is it? And this is the simple example. Let's take a look at an example from the CFILE program. This is the data type created to store the information about a single stock transaction:

```
Type
  Bucktype:Integer[8];
Stockrec=Record
   Numshare  :Integer;
   Purdate   :Daterec;   (*purchase date*)
   Optdate   :Datrec;    (*sale or option
                            close date*)
   Purtype   :Char;      (*cash, option,
                            margin etc.*)
   Cost      :Real;
   Totalval  :Bucktype;  (*this can be a big
                            number*)
   Nampnt    :Integer;   (*point to stock
                            name record*)
   Owner     :Integer;   (*pointer to owner
                            of stock*)
   Link      :Integer;   (*next transaction*)
End; (*Stockrec definition*)
```

Once we have defined two variables to be of the type Stockrec, setting one equal to another is as simple as:

```
Stock2:=Stock1;
```

The same data transfer *in a BASIC* program is going to be quite a bit more complex, especially since the elements of the record are of different types. This would prevent the programmer from using a simple

Element	**Limits**	**# bits needed**
Month	1..12	4 (0–15 max)
Day	1..31	5 (0–31 max)
Year	0..99	7 (0–127 max)
		Total bits 16

Example 2.

ADVENTURE INTERNATIONAL BOX 3435 LONGWOOD, FL 32750 (305) 862-6917

109

loop such as I used in the data example. Consider also that those elements of the Stockrec which are dates are themselves records with more than one element. When it comes to manipulating complex data records, Pascal is much more powerful than BASIC.

This same convenience also extends to arrays of variables in Pascal.

Listing 1. This is the declarations section of the CFILE program. Note that the code is almost self-explanatory. The record structures are clearly defined and can be deduced from the names of their components.

```
(* Declarations part of CFILE program *)

CONST

   Maxname   = 1000;
   Maxtrans  = 4500;
   Maxsname  = 1000;
   Namelen   = 22;
   Acctlen   = 5;
   Citylen   = 25;
   Adrlen    = 25;
   Phonelen  = 12;
   Snamelen  = 20;
   Ziplen    = 5;
   IDlen     = 5;
   SSNlen    = 11;


TYPE

   Bucktype  = Integer{8};
   Nametype  = String{Namelen};
   Acctype   = String{Acctlen};
   Adrtype   = String{Adrlen};
   Citytype  = String{Citylen};
   Ziptype   = String{Ziplen};
   Phonetype = String{Phonelen};
```

More

If we define two arrays to be of the same type such as:

MBDays:Array[1 . . 50] of Birthday;
YBDays:Array[1 . . 50] of Birthday;

then we can transfer the contents of one array to the other with the simple statement:

MBDays: = YBDays;

In the CFILE programs I don't have any occasion to use this facility, but it does make an impressive example of the abilities of Pascal.

Pascal also lets the programmer use much more meaningful names for records and elements of the records. Consider again the date example. An element of that record can be referred to as Date1.MM. It's much easier for me to think of that label as the month part of the variable Date1 than it is to make the same connection with D1%(1). And not only is it easier for me, but I'm willing to wager it would be easier for another programmer saddled with the task of maintaining the CFILE program.

Storage

Now that I've convinced you that Pascal is admirably suited for the storage and manipulation of complex

records of the type found in many business applications, I guess I should show you how efficiently you can store these records on diskettes. Another often heard, and justified, criticism of standard Pascal (as defined by Jensen and Wirth) is that the file I/O capabilities are inadequate. Since standard Pascal allows only sequential access to data files, writing data-base management programs is impossible—unless you have enough memory to read the whole file at once. UCSD Pascal has added an input/output procedure named SEEK to the Pascal system. This procedure lets the program have direct access to any record in a data file. With the addition of the file-access procedure and the addition of true text files, it is now possible to use Pascal to carry out all the operations that are possible in Applesoft BASIC with DOS 3.2. (I am purposely omitting any discussion of binary and machine-language files, as they are seldom used in business applications.)

For the moment, I'm going to confine this discussion to the use of TYPED files. These are files that are collections of a specific type of record. The file definitions for a file of friends' birthdays might look like this:

```
Type
  Birthday = record
    Name:string[25];
    Bday:Datrec; (*assume I already defined
      Datrec*)
  end;

Var
  BDfile:file of Birthday;
```

At this point we have precisely defined the structure of the Birthday file. In the same manner, I can define a file with a much more complex record structure such as:

```
Transfil:file of Stockrec;
```

A Pascal programmer can examine the type and variable definitions at the beginning of a program and deduce the exact structure of the data in a file. Have you ever tried to do the same with a BASIC program? If you're lucky, the programmer has left a good description of the file structure for you. If not, well good luck—you'll need it!

A second significant advantage of the Pascal structured file is that each record requires exactly the same amount of disk space as it does memory space. Each record is saved on diskette as a binary image of the record in memory. This is, unfortunate-

Listing 1 continued.

```
        Stattype  = (Indiv,JTROS,Partner,Corp,Trust,
                     Keogh,New,Retire);
        Stockname = String{Snamelen};
        Stockid   = String{IDlen};

        Month     = 1..12;
        Day       = 1..31;
        Year      = 0..99;

        SetofCHAR = Set of CHAR;

        Datrec    = Packed Record
          MM:Month;
          DD:Day;
          YY:Year;
        End;

        Namrec    = Record;
          Name:Nametype;
          Acctnum:Acctype;
          Tpoint,Spoint,Ppoint:integer;
        end;

        Persrec   = Record
          Address:Adrtype;
          City:Citytype;
          Zipcode:Ziptype;
          Bphone:Phonetype;
          Hphone:Phonetype;
          SSN:String{SSNlen};
          Lastact:Datrec;
          Status:Stattype;
        end;


        Stockrec  = Record
          Numshare:Integer;
          Purdate:Datrec;
          Optdate:Datrec;
          Purtype:CHAR;
          Cost:Real;
          Totalval:Bucktype;
          Nampnt:integer;
          Owner:integer;
          Link:integer;
        end;

        Transrec=Stockrec;

        Stnrec    = Record;
          ID:Stockid;
          NM:Stocknam;
        end;

        Srchkey   = Record;
          Mainkey:String{10};
          Minorkey:String{5};
          Nampoint:Integer;
        end;


     VAR

        Namefil  : File of Namrec;
        Persfil  : File of Persrec;
        Transfil : File of Transrec;
        Snamefil : File of Stnrec;

        Stnam    : Stnrec;
        Thisper  : Persrec;
        Thisnam  : Namrec;
        Thisstock: Stockrec;
        Thistrans: Transrec;

        Today    : Datrec;


        Astring,Bstring : String;
        Xlin,Ylin,Nrec,I,J,K,Recnum,Nextrec: Integer;

        CH,Selection: CHAR;
        OKset:Set of CHAR;

        SNarray:Array{1..Maxsname} of Stockid;
```

ly, not true in Applesoft BASIC (other versions of BASIC do allow the storage of data in memory image form). With Applesoft a date record consisting of three integers, MM%, DD% and YY%, would require six bytes of memory space but might require as much as nine bytes of diskette space since it would be written out as 12(CR)31(CR)80(CR). In Applesoft, each numeric item is actually written to the disk as a string, and a delimiter (usually a carriage return) is needed to separate the data items. The ability to store data records on diskette in memory image format can more than double the amount of data which can be saved in a given amount of file space. You should also remember that the Pascal diskette format allows about 30 percent more storage space per diskette than DOS 3.2.

There are five basic operations on typed files:

- *RESET*(BDfile,'MarkDisk:MarkBDays') opens a data file where BDfile is a file which has been defined in the variable declarations of the programs and MarkBDays is the name of a file on the diskette MarkDisk. (In the Pascal operating system, each diskette is given a volume name.)
- *SEEK*(BDfile,Recnum) positions a file access pointer to the record number defined by Recnum (Recnum must be an integer). This addition to standard Pascal allows random access to data files.
- *GET*(BDfile) transfers the record pointed to by the access pointer to a special variable of the type Birthday which is called the Window Variable. You can then manipulate this variable just like any other Birthday record.
- *PUT*(BDfile) transfers the contents of the window variable to the disk file record pointed to by the access pointer.
- *CLOSE*(BDfile) closes the file and makes sure that any changes in the file data get transferred from the file buffer in memory to the diskette.

In a business program, these file manipulation commands are generally incorporated into procedures which check to see if the required data diskette is in one of the disk drives and prompt the user if the diskette is not available. These file manipulation commands allow the Pascal programmer to set up sophisticated file structures to exactly suit the needs of a given program. For example, the CFILE programs use the following file structures:

- *NAMEFIL:FILE OF NAMREC;* This file contains client names and pointers to record numbers in the personal data file and the transaction file.
- *PERSFIL:FILE OF PERSREC;* This is the file which holds the personal data on the clients.
- *TRANSFIL:FILE OF STOCKREC;* This is the file of transactions. The file is a series of linked-list structures. The name record for each client contains a pointer to the first transaction in the client's list.
- *SNAMEFIL:FILE OF STNREC;* Stock names and their abbreviations are kept in this file.

The Name file can be thought of as a key file. When searching for data pertaining to a particular client, the program searches this file (or an image of it in memory) for a match to the required name. When a match is found, the location of the personal data record and the start of the transaction list are taken from the client's Name record. The transactions are saved in a linked-list structure so that each client may have a different number of associated transactions without requiring enough disk space to save the maximum number of transactions for all the clients. The details of linked-list file structures and their advantages and disadvantages are beyond the scope of this article.

The UCSD Pascal system also lets the programmer regard any data file as an unstructured set of blocks of 512 bytes each. This ability is particularly useful when you want to copy the entire contents of one file into another file without worrying about the exact contents of the first (this is known as backing up the data file).

Since the program does not need to break the file blocks up into individual records, the read and write procedures are done much more rapidly. In fact, it is possible to back up the full transactions diskette in about 1.5 minutes. Copying the files record-by-record requires about five times as long.

The backup procedure used in BASIC programs generally requires you to initialize and copy a complete diskette. This may require as many as five minutes per diskette. In addition, the only way to copy a given file

NEW! TPM* for TRS-80 Model II
NEW! System/6 Package
Computer Design Labs

Z80* Disk Software

We have acquired the rights to all TDL software (& hardware). TDL software has long had the reputation of being the best in the industry. Computer Design Labs will continue to maintain, evolve and add to this superior line of quality software.

— Carl Galletti and Roger Amidon, owners.

Software with Manual/Manual Alone

All of the software below is available on any of the following media for operation with a Z80 CPU using the CP/M* or similar type disk operating system (such as our own TPM*).

for TRS-80* CP/M (Model I or II)
for 8" CP/M (soft sectored single density)
for 5¼" CP/M (soft sectored single density)
for 5¼" North Star CP/M (single density)
for 5¼" North Star CP/M (double density)

BASIC I

A powerful and fast Z80 Basic interpreter with EDIT, RENUMBER, TRACE, PRINT USING, assembly language subroutine CALL, LOADGO for "chaining", COPY to move text, EXCHANGE, KILL, LINE INPUT, error intercept, sequential file handling in both ASCII and binary formats, and much, much more. It runs in a little over 12 K. An excellent choice for games since the precision was limited to 7 digits in order to make it one of the fastest around. $49.95/$15.

BASIC II

Basic I but with 12 digit precision to make its power available to the business world with only a slight sacrifice in speed. Still runs faster than most other Basics (even those with much less precision). $99.95/$15.

BUSINESS BASIC

The most powerful Basic for business applications. It adds to Basic II with random or sequential disk files in either fixed or variable record lengths, simultaneous access to multiple disk files, PRIVACY command to prohibit user access to source code, global editing, added math functions, and disk file maintenance capability without leaving Basic (list, rename, or delete). $179.95/$25.

ZEDIT

A character oriented text editor with 26 commands and "macro" capability for stringing multiple commands together. Included are a complete array of character move, add, delete, and display function. $49.95./$15.

ZTEL

Z80 Text Editing Language - Not just a text editor. Actually a language which allows you to edit text and also write, save, and recall programs which manipulate text. Commands include conditional branching, subroutine calls, iteration, block move, expression evaluation, and much more. Contains 36 value registers and 10 text registers. Be creative! Manipulate text with commands *you write using Ztel.* $79.95/$25.

TOP

A Z80 Text Output Processor which will do text formatting for manuals, documents, and other word processing jobs. Works with any text editor. Does justification, page numbering and headings, spacing, centering, and much more! $79.95/$25.

MACRO I

A macro assembler which will generate relocateable or absolute code for the 8080 or Z80 using standard Intel mnemonics plus TDL/Z80 extensions. Functions include 14 conditionals, 16 listing controls, 54 pseudo-ops, 11 arithmetic/logical operations, local and global symbols, chaining files, linking capability with optional linker, and recursive/reiterative macros. This assembler is so powerful you'll think it is doing all the work for you. It actually makes assembly language programming much less of an effort and more creative. $79.95/$20.

MACRO II

Expands upon Macro I's linking capability (which is useful but somewhat limited) thereby being able to take *full* advantage of the optional Linker. Also a time and date function has been added and the listing capability improved. $99.95/$25.

LINKER

How many times have you written the same subroutine in each new program? Top notch professional programmers compile a library of these subroutines and use a Linker to tie them together at assembly time. Development time is thus drastically reduced and becomes comparable to writing in a high level language but with all the speed of assembly language. So, get the new CDL Linker and start writing programs in a fraction of the time it took before. Linker is compatible with Macro I & II as well as TDL/Xitan assemblers version 2.0 or later. $79.95/$20.

DEBUG I

Many programmers give up on writing in assembly language even though they know their programs would be faster and more powerful. To them assembly language seems difficult to understand and follow, as well as being a nightmare to debug. Well, not with proper tools like Debug I. With Debug I you can easily follow the flow of any Z80 or 8080 program. Trace the program one step at a time or 10 steps or whatever you like. At each step you will be able to see the instruction executed and what it did. If desired, modifications can then be made before continuing. It's all under your control. You can even skip displaying a subroutine call and up to seven breakpoints can be set during execution. Use of Debug I can pay for itself many times over by saving you valuable debugging time. $79.95/$20.

DEBUG II

This is an expanded debugger which has all of the features of Debug I plus many more. You can "trap" (i.e. trace a program until a set of register, flag, and/or memory conditions occur). Also, instructions may be entered and executed immediately. This makes it easy to learn new instructions by examining registers/memory before and after. And a RADIX function allows changing between ASCII, binary, decimal, hex, octal, signed decimal, or split octal. All these features and more add up to give you a very powerful development tool. Both Debug I and II must run on a Z80 but will debug both Z80 and 8080 code. $99.95/$20.

ZAPPLE

A Z80 executive and debug monitor. Capable of search, ASCII put and display, read and write to I/O ports, hex math, breakpoint, execute, move, fill, display, read and write in Intel or binary format tape, and more! on disk

APPLE

8080 version of Zapple

NEW! TPM now available for TRS-80 Model II!

TPM*

A NEW Z80 disk operation system! This is not CP/M*. It's better! You can still run any program which runs with CP/M* but unlike CP/M* this operating system was written specifically for the Z80* and takes full advantage of its extra powerful instruction set. In other words its not warmed over 8080 code! Available for TRS-80* (Model I or II). Tarbell, Xitan DDDC, SD Sales "VERSA-FLOPPY", North Star (SD&DD), and Digital (Micro) Systems. $79.95/$25.

SYSTEM MONITOR BOARD (SMB II)

A complete I/O board for S-100 systems. 2 serial ports, 2 parallel ports, 1200/2400 baud cassette tape interface, sockets for 2K of RAM, 3-2708/2716 EPROM's or ROM, jump on reset circuitry. Bare board $49.95/$20.

ROM FOR SMB II

2KX8 masked ROM of Zapple monitor. Includes source listing $34.95/$15.

PAYROLL (source code only)

The Osborne package. Requires C Basic 2.
5" disks $124.95 (manual not included)
8" disks $ 99.95 (manual not included)
Manual $20.00

ACCOUNTS PAYABLE/RECEIVABLE (source code only)

By Osborne, Requires C Basic 2
5" disks $124.95 (manual not included)
8" $99.95 (manual not included)
Manual $20.00

GENERAL LEDGER (source code only)

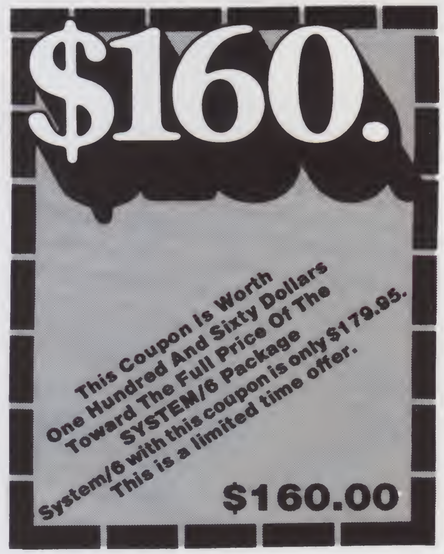
By Osborne. Requires C Basic 2
5" disks $99.95 (manual not included)
8" disks $99.95 (manual not included)
Manual $20.00

C BASIC 2

Required for Osborne software. $99.95/$20.

SYSTEM/6

TPM with utilities, Basic I interpreter, Basic E compiler, Macro I assembler, Debug I debugger, and ZEDIT text editor.
Above purchased separately costs $339.75
Special introductory offer: Only $179.75 with coupon!!



ORDERING INFORMATION

Visa, Master Charge and C.O.D. O.K. To order call or write with the following information.

1. Name of Product (e.g. Macro I)
2. Media (e.g. 8" CP/M)
3. Price and method of payment (e.g. C.O.D.) include credit card info. if applicable.
4. Name, Address and Phone number.
5. For TPM orders only: Indicate if for TRS 80, Tarbell, Xitan DDDC, SD Sales (5¼" or 8"). ICOM (5¼" or 8"), North Star (single or double density) or Digital (Micro) Systems.
6. N.J. residents add 5% sales tax.

Manual cost applicable against price of subsequent software purchase in any item except for the Osborne software.

For information and tech queries call
609-599-2146

For phone orders ONLY call toll free
1-800-327-9191
Ext. 676
(Except Florida)

OEMS

Many CDL products are available for licensing to OEMs. Write to Carl Galletti with your requirements.

* Z80 is a trademark of Zilog
* TRS-80 is a trademark for Radio Shack
* TPM is a trademark of Computer Design Labs. It is not CP/M*
* CP/M is a trademark of Digital Research

Prices and specifications subject to change without notice.

DEALER INQUIRIES INVITED.

COMPUTER DESIGN LABS

✓18

342 Columbus Avenue
Trenton, N.J. 08629

without destroying others on the backup diskette is to copy the file record-by-record. Believe me, this can get slow!

Conclusions

If I haven't convinced you by now that there are some real advantages to writing microcomputer business programs in Pascal, I'm going to consider you a reactionary BASIC diehard. This puts you in nearly the same category as the IBM salesman who was rumored to have said, "Those PDP-11s are just toys. You'll never be able to do any serious business computing with those minicomputers!"

For those of you who are only skeptical or who skimmed the article, I'll reiterate the main points. If you are considering business software for a microcomputer, give Pascal a chance. It has some significant advantages over BASIC:

1. Structured programming, which leads to more precise definition of data types, programming approaches and software solutions to your business problems.
2. Intelligible source code, the best way to guarantee your program can be maintained when your programmer decides he'd rather write magazine articles.
3. Efficient use of memory resources for the storage of data. No computer ever has enough memory. With Pascal you can store more data in less. With luck, it will take you another year to outgrow your computer.
4. The ability to manipulate grouped data items as a single record. The less time your programmer spends writing code to move data, the more time he can spend making it easier for you to get information into and out of the computer.
5. More efficient ways to store data on your diskette (see 3).
6. Programs which run more quickly. If your program was written in Pascal, you'd only have time to skim this article while the customer file was sorting. If you're still using BASIC, go back to the beginning and read the article again. You'll have time! ■

References

Problem Solving Using Pascal by Kenneth L. Bowles. A good introduction, but the emphasis is on examples suited to college students.

PASCAL User Manual and Report by Kathleen Jensen and Niklaus Wirth. The authoritative definition of the Pascal language. Doesn't mention UCSD extensions to language. Give this one to your favorite programmer. (This one and *Problem Solving* come with the Apple Pascal package.)

Programming in Pascal by Peter Grogono. Slightly more advanced than Bowles' book. Several excellent examples of linked list structures and I/O procedures. I particularly liked the part about record structures.

Algorithms + Data Structures = Programs by Niklaus Wirth. A much more advanced work. For programmers only. Good sections on sets and sorting arrays. About 150 pages on tree structures and sentence parsing, which I haven't found any use for in business programs.

Beginner's Guide for the UCSD Pascal System by Kenneth Bowles. If you want to convert a programmer, buy this book, the accompanying program disk and an Apple Pascal system. The book assumes only that you can read, have a system to work with and want to learn.

REAL TIME SAVER

Minutes are money. So, when it's possible to easily handle many times your workload, with better results, you're on to something.

And, that something is Programma's versatile and powerful Word Processing System. Because it's designed from the user's viewpoint, it's easy to work with. You can start right out doing basic word processing. And, since each step logically leads to the next, you'll be handling even complex work problems in a very short time.

Our WPS consists of two superior programs, Apple PIE (Programma Improved Editor) and FORMAT. PIE is a free-form, live-screen-oriented editor, for creating and editing text for processing. The commands let you enter or alter anything, anywhere on your screen. You can search and replace, add, delete, move, or insert, by character, word, line or paragraph. And you know what's happening, as you see it right on the screen. PIE also allows use of a lower case adapter, and can even be used as a software development tool.

The other half of our real time saver is FORMAT. It uses simple, embedded codes to set formats for letters, manuals, scripts, documents or catalogs to your specifications.

You want this centered?...that underlined?... those areas indented?...something paragraphed? You make the decisions, FORMAT does the work immediately, accurately. Then, if before printing, you want to revise copy, or change a document's final appearance, you can do it with ease. FORMAT is very forgiving!

To make the most of your time, order today. Our WPS for the 32K Apple II, on mini-floppies, with thorough documentation, is now just $129.95. At your local computer store. Or direct from Programma, 2908 No. Naomi Street, Burbank, CA 91504. (213) 954-0240.

PROGRAMMA'S WORD PROCESSING SOFTWARE FOR THE APPLE II*

PROGRAMMA
INTERNATIONAL, INC

✓305

* Apple II is a trademark of Apple Computer, Inc.

✓Reader Service—see page 210

The author lifts the veil of mystery covering heretofore undocumented Z-80 instructions.

Secret Codes Revealed

By Edwin E. Freed

The Z-80 microprocessor was originally designed as an 8080 enhancement. In addition to running faster, it added a number of new features, and yet maintained machine-code compatibility with the 8080. This meant that only unused operation codes on the 8080 could be expanded to implement these features.

The purpose of this article is to describe how these codes were used. Undocumented codes have been found which you can use to advantage. Manipulation of the halves of the IX and IY registers, additional rotate instructions and some duplication of instructions have also been uncovered.

Some of the operation codes found here are not officially documented or supported by the manufacturer; such codes are underlined in the tables. These undocumented operations have been tested and found to work on Zilog Z-80 and Z-80A chips. However, there is no guarantee that these codes will continue to function on future generations of the Z-80.

On any eight-bit processor, there are 256 possible first bytes for instructions. On the 8080, all but 12 of these are used. In valid 8080 instructions, there are no undocumented codes that result from values in successive bytes of the instructions. To maintain 8080 software compatibility, all of the Z-80's additional features use these 12 codes. Some of the codes have been expanded to as many as four-byte instructions.

Eight of the 12 available op codes implement a single operation (Table 1). These are the relative jump instructions and the register bank exchange operation, which could have been packed into a single op code, of course, but only at the expense of adding an extra byte to the instructions.

To be useful, the relative jump instructions must be two-byte. A three-byte relative jump would offer no space savings over the traditional absolute jump of the 8080. The only gain would be some relocatability of the machine code.

The register bank exchange instructions manipulate the Z-80's alternate register set. Since these instructions are primarily intended for fast interrupt processing, it is crucial that they be one-byte in order to execute as quickly as possible.

Hex code		Instruction	
08		EX	AF, AF'
10	ee	DJNZ	E
18	ee	JR	E
20	ee	JR	NZ, E
28	ee	JR	Z, E
30	ee	JR	NC, E
38	ee	JR	C, E
D9		EXX	

Table 1. The Z-80's relative jump and bank exchange instructions.

In the implementation of the Z-80, additional 16-bit index registers IX and IY were added. Two operation codes, DD and FD, provide instruction set support. The designers of the Z-80 added the new index registers simply: the IX and IY instructions consist of DD or FD followed by an instruction which refers the HL register pair. For example, the code to load HL immediate (LD HL,nn) is 2A nn; the instruction LD IX,nn is DD 2A nn. The equivalent IY instruction is FD 2A nn. In either case, the two-byte operand nn immediately follows the instruction.

When an instruction that uses HL as an indirect memory pointer is prefaced with DD or FD, something different happens. An additional byte is required in the instruction. The value of this byte, called the displacement, is added to the IX or IY register. This

```
0000   24     INC   H      ;Increment H
0001   DD24   INC   HX     ;Increment top half of IX
0003   2D     DEC   L      ;Decrement L
0004   FD2D   DEC   LY     ;Decrement bottom half of IY
0006   8C     ADC   A,H    ;Add with carry H to A
0007   FD8C   ADC   A,HY   ;Add with carry HY to A
0009   AD     XOR   A,L    ;. . .
000A   DDAD   XOR   A,LX
000C   B5     OR    A,L
000D   FDB5   OR    A,LY
```

Listing 1. A sampling of instructions referencing IX and IY as four eight-bit registers. The format is that of a typical assembly-language listing. First, an instruction which operates on H or L is shown, then a analogous instruction for an index register.

Address correspondence to Edwin E. Freed, Platt Campus Center, Harvey Mudd College, Claremont, CA 91711.

High order 5 bits of second byte

Low order 3 bits	00	08	10	18	20	28
0	RLC B	RRC B	RL B	RR B	SLA B	SRA B
1	RLC C	RRC C	RL C	RR C	SLA C	SRA C
2	RLC D	RRC D	RL D	RR D	SLA D	SRA D
3	RLC E	RRC E	RL E	RR E	SLA E	SRA E
4	RLC H	RRC H	RL H	RR H	SLA H	SRA H
5	RLC L	RRC L	RL L	RR L	SLA L	SRA L
6	RLC (HL)	RRC (HL)	RL (HL)	RR (HL)	SLA (HL)	SRA (HL)
7	RLC A	RRC A	RL A	RR A	SLA A	SRA A

Low order 3 bits	30	38	40–7F	80–9F	A0 – FF
0	RLO B	SRL B	BIT b,B	RES b,B	SET b,B
1	RLO C	SRL C	BIT b,C	RES b,C	SET b,C
2	RLO D	SRL D	BIT b,D	RES b,D	SET b,D
3	RLO E	SRL E	BIT b,E	RES b,E	SET b,E
4	RLO H	SRL H	BIT b,H	RES b,H	SET b,H
5	RLO L	SRL L	BIT b,L	RES b,L	SET b,L
6	RLO (HL)	SRL (HL)	BIT b,(HL)	RES b,(HL)	SET b,(HL)
7	RLO A	SRL A	BIT b,A	RES b,A	SET b,A

Table 2. The shift and rotate instructions. The first byte is always CB. Previously undocumented left rotate instructions are underlined. In the bit manipulation instructions, the affected bit b is indicated by bits 3, 4 and 5 in the second byte.

modified value is then used as a memory pointer to obtain the operand.

This technique for using the index register instructions leaves many questions unanswered. For instance, what happens when DD or FD is followed by an instruction which refers to the BC or DE, not the HL register pair? When tested, these instructions functioned normally and acted on the original (not IX, IY) register pair involved. So there are actually many *duplicate instructions* available, all useless because they are one byte longer than the original instruction.

Prefixing an instruction that references the H or L register with DD or FD can lead to some useful results, however. You can manipulate the high and low halves of the IX and IY registers as easily as any other eight-bit register! (See Listing 1.) If you use an instruction that references as another eight-bit register, like B or E, it behaves normally.

No new assembly-language instruction mnemonics are needed to allow access to these new instructions. New names which refer to the IZ and IY registers as four eight-bit registers are used. The chosen names are HX, LX, HY and LY. HX refers to the high-order eight bits of IX, LX refers to the low-order eight bits, and so on.

The CB op code group is used to implement the individual register rotate and bit manipulation instructions. Their format is always two bytes, with the first byte being CB, and the second byte indicating both operation and the register affected.

In the decomposition of the CB group (Table 2), only eight instructions were left undefined. Upon testing, these turned out to be a hitherto undocumented form of a left rotate instruction that shifts a one into the low-order bit and moves the high-order bit into the carry flag. This, in effect, performs a 2n + 1 operation on an eight-bit register. The chosen op code for this new function is RLO (ro-

High order 5 bits of second byte

Low order 3 bits	80	88	90	98	A0	A8	B0	B8
0	NOP	NOP	NOP	NOP	LDI	LDD	LDIR	LDDR
1	NOP	NOP	NOP	NOP	CPI	CPD	CPRI	CPDR
2	NOP	NOP	NOP	NOP	INI	IND	INIR	INDR
3	NOP	NOP	NOP	NOP	OUTI	OUTD	OTIR	OTDR
4–7	NOP	NOP	NOP	NOP	NOP	NOP	NOP	NOP

Table 3. The block move/transfer instructions. The first byte of all these instructions is ED.

Reader Service—see page 210

tate left, shift in one). The format for this new rotate instruction is exactly the same as for any other in the family.

The ED op code group is used to implement a multitude of Z-80 features. All the block transfer, block search and additional interrupt control functions are included in this group. All instructions consist of ED followed by a single byte to indicate the exact operation. No additional arguments are used. Although 256 new instructions could have been implemented, only 60 of these are actually defined by Zilog.

It happens that none of the undefined codes in this group do anything useful. The undefined codes 00–3F and C0–FF act as NOPs. Codes in the range 80–BF are used for the block control instructions (Table 3). The remaining codes in the range 40–7F perform a variety of functions. The undefined ones in this group tend to be duplicates of existing codes (Table 4).

It should be noted that prefixing instructions in the ED group with DD or FD will not create a corresponding IX or IY operation. This is regrettable, but the ED group appears to be the only one so affected.

Conclusion

It should be possible to modify most Z-80 assembler to accept the foregoing undocumented instructions, if a source to the assembler is available. If not, you can insert the hex codes for the instructions inline as data statements. Code the new index register instructions as DD or FD, followed by the appropriate H or L referencing instruction. ■

	High order 5 bits of second byte															
Low order 3 bits	40		48		50		58		60		68		70		78	
0	IN	B,(C)	IN	C,(C)	IN	D,(C)	IN	E,(C)	IN	H,(C)	IN	L,(C)	IN	(HL),(C)	IN	A,(C)
1	OUT	(C),B	OUT	(C),C	OUT	(C),D	OUT	(C),E	OUT	(C),H	OUT	(C),L	OUT	(C),(HL)	OUT	(C),A
2	SBC	HL,BC	ADC	HL,BC	SBC	HL,DE	ADC	HL,DE	SBC	HL,HL	ADC	HL,HL	SBC	HL,SP	ADC	HL,SP
3	LD	(nn),BC	LD	BC,(nn)	LD	(nn),DE	LD	DE,(nn)	LD	(nn),HL	LD	HL,(nn)	LD	(nn),SP	LD	SP,(nn)
4	NEG		NEG		NEG		NEG		NEG		NEG		NEG		NEG	
5	RETN		RETI		RETN		RETI		RETN		RETI		RETN		RETI	
6	IM	0	NOP		IM	1	IM	2	NOP		NOP		NOP		NOP	
7	LD	I,A	LD	R,A	LD	A,I	LD	A,R	RRD		RLD		NOP		NOP	

Table 4. Miscellaneous instructions with ED as the first byte. Note that the NOPs in the same row as the IM instructions may have hitherto undiscovered effects on the interrupt system.

ORDER ENTRY/ACCOUNTS RECEIVABLE SYSTEMS

Fyrnetics President, Larry Larsen, and Controller, Dennis Turek, considering some of the features of their new security system.

THE COMPANY:
FYRNETICS, INC.
1021 DAVIS ROAD
ELGIN, ILLINOIS 60120
312-742-0282

Fyrnetics, Inc. of Elgin, Illinois has grown from a private manufacturer of ionization smoke detectors for companies, such as Sears & Roebuck, to a full line manufacturer of wireless security products which are sold through a worldwide network of dealers and distributors. Since saturation of the residential security market is estimated to be less than 3% in the United States, their major marketing effort has been directed toward distribution through dealers who demonstrate and sell electronic products to the consumer. This marketing approach generated such a high volume of sales orders and invoices that it became necessary to install a computer system in 1977 in order to handle the multi-million dollar annual sales volume.

"Fyrnetics initially installed a Wang 2200 computer system, equipped with seven CRT terminals, four 10 megabyte hard disk drives, and two line printers at a monthly lease cost of over $4000.00. By late 1979, the company had invested nearly six man-years in our own software development", comments Fyrnetics president, Larry Larsen **"As a part of a cost cutting program, we decided to replace the Wang** equipment with a microcomputer system. We reviewed all available micro systems and concluded that the MSI hardware and software packages were best suited to our needs."

The modern production facility at Fyrnetics, Inc. where wireless home security products are designed and manufactured. The company also has production facilities in Hong Kong where larger quantities of their products are manufactured.

"We were particularly pleased with the system-generation capability of the MSI business packages, which allowed us to utilize our existing continuous form sales orders, invoices, and packing lists. The requirements for our computer system were rather demanding since we had large customers such as Montgomery Wards and Wicks, each having up to 300 stores. Each store is treated as an individual customer during the order entry and shipping process. However, payment is made from a central accounting office with many stores on a single check. Our system had to allow us to properly credit the payment to many different store locations and invoices. This feature was a part of the MSI Accounts Receivable software package. **Our accounts receivable system handles over 750 regular customers with over 3000 open invoices and 10,000 transactions per month for us."**

The LIFESAVER line of wireless home security products manufactured by Fyrnetics, Inc. Anyone desiring more information on this interesting product line should contact Fyrnetics, Inc. at the above address.

CONSIDERATIONS FOR THE SELECTION OF ACCOUNTS RECEIVABLE SYSTEMS

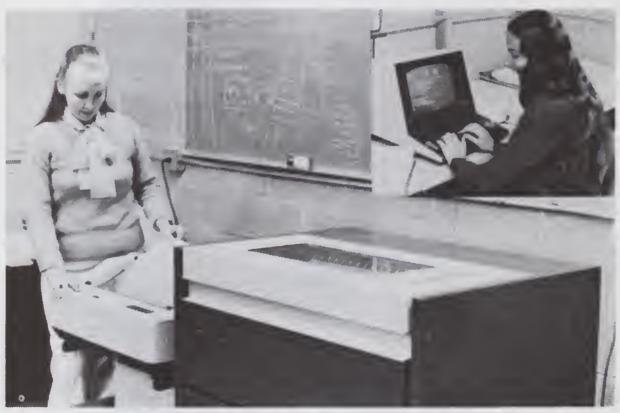
The selection of business computer systems today involves the careful consideration of many different factors. Even though the cost of computer systems has dropped substantially, we considered the selection process to be highly critical to us because of the tremendous need for a highly reliable computer system in our daily operations. Due to our high volume of sales transactions we were highly dependent upon the system for order processing and for information. We considered the following issues to be key to our selection of the MSI system:

LARGE DATA BASE - The processing of over 700 orders per month, with 3000 open invoices and 5000 active statement items required that we have easy and efficient on-line access to our large data base. The MSI system provides a large selection of data reports for open orders, backorders, invoices, credit memos, as well as customer statements and account status information.

EFFICIENT PAPERWORK FLOW - The processing of our large volume of sales orders required an efficient system for printing sales orders, packing lists, invoices, and customer statements. The MSI system offers a convenient system generation program which allows the use of any desired format for pre-printed continuous forms. In addition, packing lists and customer invoices are generated automatically as sales orders are processed.

GENERAL LEDGER TIE-IN - Due to the large volume of individual invoices and cash receipts, we required an automatic posting procedure for our general ledger programs in order to minimize the data entry process. The MSI system offers a complete general ledger program package which links automatically to the other business program modules. All invoices, as well as cash receipts, are automatically written to the general ledger posting files from which individual journals are created. This procedure insures the generation of balanced journals and greatly reduces the time requirement for generation of monthly income statement and balance sheets.

SYSTEM INTEGRATION - The MSI system is fully integrated. The order entry system is linked to inventory for correct pricing, description of items on order. The inventory system is also linked to general ledger to allow different categories of products sold to be automatically posted to the correct sales accounts. The MSI inventory system provides complete cost accounting information for both labor and material. **The MSI programs provide the big machine capability that we need and yet provide the flexiblilty that we desire.**



The MSI computer system drives two line printers at Fyrnetics, Inc.

The MSI computer system at Fyrnetics, Inc. employs 10 megabyte hard disk drives to contain the large on-line data base.

SUPPORT - The availability of source listings for all of the MSI business software was an added incentive to select the MSI system. This has allowed us to make some specialized enhancements to our programs easily. **MSI really delivered for us allowing the replacement of an expensive WANG 2200 system with a comparable MSI system at a fraction of the cost.**

If you would like to have more information on MSI business computer systems, call or write, **Midwest Scientific Instruments, Inc., 220 W. Cedar, Olathe, KS 66061, 800-255-6638, Telex 42525(MSI A OLAT).**

MSI ..... Helping to make your business run better.

Encipher and decipher secret messages with this Playfair cipher program for the Apple II.

Crack That Code

By Dr. Richard C. Vile, Jr.

Few people have not at one time or another played with secret messages. Codes and ciphers have been a source of fascination for many centuries. History is filled with many enthralling stories which have their basis in cryptography.

Cryptography deals with secret messages. Of the various types, ciphers and codes are both produced by transforming the original message, or plain text. Ciphers are produced by transformations which operate on single letters or small groups of letters at a time according to some relatively simple algorithmic process. Codes are produced by transformations which usually operate on entire words and sometimes entire sentences at a time. Codes may require elaborate apparatus in the form of code books, and are usually more vulnerable to theft than to analysis. Plain text is transformed into cipher text, the secret message, by the process of encipherment, the application of one or more codes or ciphers. Cipher text is rendered understandable by the inverse process of decipherment.

The term cryptography refers to the processes of enciphering and deciphering messages. Cryptanalysis is the study of the methods used to crack or solve codes and ciphers when the enciphering algorithm is not known. Cryptology refers to the combined sciences of cryptography and cryptanalysis and the study of related techniques.

Most cryptographic techniques rely on a general system and a specific key. The general system is the overall method employed, such as simple substitution, columnar transposition, and so on.

The Playfair cipher is an example of a general system that uses digraphic substitution. A specific key consists of the particular details of the general system, which are varied from one application of the system to the next. The Playfair cipher uses a specific key in the form of a key word or key phrase. The key is used to build a so-called Playfair square, which is a 5×5 rectangular table used in the cryptographic process.

The Playfair Cipher

This system was invented by the famous scientist Sir Charles Wheatstone, but was popularized by his friend and ·fellow scientist Baron Playfair. It was used by the British Army for many years as its field cipher, the technique used by soldiers in the field. It is elegant and simple to use, but more difficult to crack than simple monographic (single letter) substitution ciphers.

A Playfair square is a 5×5 square table filled with the letters of the alphabet, except for J (I is substituted for J). Any agreed-upon method of filling in the letters may be used—the only one to avoid is, of course, ordinary alphabetical order. The most common technique is to choose a key word or phrase—for example "MICROCOMPUTERS ARE FUN"—and first fill in the initial part of the square with the unique letters of that key (see Fig. 1). The remaining cells in the square are then filled in with the letters unused in the key, in alphabetical order (see Fig. 2). A good key will have from ten to 15 distinct letters, although any key phrase desired will work.

A message which is to be enciphered using the Playfair square is first split up into successive pairs, or digraphs. Pairs with identical letters are avoided by the insertion of an extra letter called a null. The null may be any letter mutually agreed upon in advance. For example, the plain text "The APPLE II Computer System" would be split up as follows (we capitalize for uniformity):

TH EA PX PL EI IC OM PU TE RS YS TE MX

The null letter X was used in this example. Notice the null at the end of the message, to make an even number of letters in all.

Each plain-text digraph is transformed into a cipher text digraph using the Playfair square. The specific transformation applied to the plain-text digraph depends on the relative position of the letters of the digraph in the Playfair square. There are three distinct possibilities:

- The two letters are in the same row of the square.
- The two letters are in the same col-

Dr. Richard C. Vile, Jr., 3467 Yellowstone Drive, Ann Arbor, MI 48105.

TH→UK	EP→SU	LA→GB	YF→WB	AI→FM
RC→OR	IP→MU	HE→LU	RW→IY	AS→DP
AC→NM	TU→ET	AL→BG	LY→YR	IN→CF
VE→YP	NT→KN	ED→SB	BY→LR	SI→UO
RC→OR	HA→GF	RL→EY	ES→SP	WH→IW
EA→PB	TS→EP	TO→SC	NE→BT	

Table 1.

KXJEY UREBE ZWEHE WRYTU HEYFS KREHE GOYAI WTTTU OLKSY CAJPO BOTEI ZONTX BYBWT GONEY CUZWR GDSON SXBOU YWRHE BAAHY USEDQ

Table 2.

umn of the square.

- The two letters are in different rows and columns.

Same row. Replace each plain-text letter of the pair with the one to its immediate right in the same row of the Playfair square. Imagine the row to be circular, so that the rightmost letter of a row will cipher to the first letter of the same row.

Same column. Replace each plain-text letter of the digraph by the one directly below it in the same column of the Playfair square. Again, imagine that the columns are circular, so that the bottom letter of each column will cipher to the top letter of the same column.

Different row and column. The two letters of the digraph form the opposite corners of a rectangle within the Playfair square. Each letter of the pair is replaced by the letter in the same row of the Playfair square which is one of the other two corners of the rectangle formed.

The methods used to decipher a message are simply the inverse of those just described. Thus, if a cipher-text pair belongs to the same column of the square, it is deciphered by replacing each letter with the one directly above it in the same column, the *top letter* of any column being replaced by the bottom letter of the same column.

Examples and Exercises

Using the Playfair square constructed in Fig. 2, now consider the following plain text: The Playfair cipher was actually invented by Sir Charles Wheatstone. The cipher text produced from this message is shown

KEY: MICROCOMPUTERS ARE FUN

M	I	C	R	O
P	U	T	E	S
A	F	N		

Fig. 1. Filling in the KEY letters.

KEY: MICROCOMPUTERS ARE FUN

M	I	C	R	O
P	U	T	E	S
A	F	N	B	D
G	H	K	L	Q
V	W	X	Y	Z

Fig. 2. Completing the Playfair Square.

in Table 1, one digraph at a time:

The circled examples are worked out in detail in Fig. 3.

To enhance your understanding of the Playfair cipher, try the following exercises:

a. Describe precisely the three methods of deciphering individual digraphs.

b. Construct the Playfair square for the following key: ROYAL NEW ZEALAND NAVY.

c. Using the square constructed in b, decipher the message in Table 2 (sent 9:30 AM, August 7, 1943, to report the sinking of John F. Kennedy's PT109). Note that nulls were not used to eliminate double letters in the Table 2 cipher text. In addition, some of the letters may have been garbled in transmission.

Apple II Playfair Program

Listing 1 presents an Applesoft pro-

gram that facilitates the use of the Playfair cipher. The user may construct Playfair squares and both encipher and decipher messages. Input and output is quite flexible and allows the following combinations:

Input—Keyboard, file.

Output—Keyboard only; file only; keyboard and file; keyboard and printer; file and printer; keyboard, file and printer.

These options may be dynamically changed during a given program run. The output, whether cipher text or plain text, is formatted in standard cryptographic fashion: groups of five letters, separated by a space.

Program Notes

Fig. 4 shows the relationships between the various routines of the program. Arrows in the diagram point from a calling routine to a called routine. The numbers above each oval represent the line number at which the corresponding routine begins. Thus, the Gather routine at line 1000 calls the Scan routine at line 2000.

Even though the program is lengthy, a study of the following comments on each routine together with Fig. 4 should suffice to easily understand its operation. Incidentally, the program of Listing 2 presents a condensed version of these notes and should be the program invoked by letter [A] in the master menu of the Playfair program (READ INTERNALS DOCUMENTATION).

1–499—PRELIMS. A couple of one-time-only initializations are performed. Arrays are dimensioned, the function FN M6(A) is defined, and the routines COUT-POKES and INTRO are invoked.

Listing 1. Playfair Cipher program.

```
1   REM ==========================
2   REM =                        =
3   REM =    PLAYFAIR CIPHER     =
4   REM =                        =
5   REM =           BY           =
6   REM =DR. RICHARD C. VILE, JR.=
7   REM =                        =
8   REM =                        =
9   REM ==========================
14  DIM PF$(4,4),US(26),ROW(26),COL(26)
15  DEF  FN M6(A) =  INT ((A / 6 -  INT (A / 6)) * 6 + .05) *  SGN
     (A / 6)
20 D$ =  CHR$ (4)
21 HC = 0: REM THIS CANNOT GO IN NORMAL INITIALIZATION ROUTINE
100  GOSUB 9000: REM  INTRODUCTION
105  GOSUB 6000: REM  SETUP COUT INTERFACE
500  REM MENU ROUTINES
501  REM ============
505  GOSUB 1800: REM  INITIALIZE
506  HOME : GOSUB 2700
508  NORMAL : VTAB 5: HTAB 5
510  PRINT "CHOOSE ONE OF THE FOLLOWING:": PRINT
515  NORMAL : HTAB 5
520  POKE 249,219: CALL 250: PRINT "A] READ INTERNALS DOCUMENTATIO
     N"
521  HTAB 5: CALL 250: PRINT "B] ENCIPHER TEXT"
522  HTAB 5: CALL 250: PRINT "C] DECIPHER TEXT"
523  HTAB 5: CALL 250: PRINT "D] BUILD A NEW PLAYFAIR SQUARE"
524  HTAB 5: CALL 250: PRINT "E] DISPLAY CURRENT PLAYFAIR SQUARE"
525  HTAB 5: CALL 250: PRINT "F] TURN ON PRINTER OUTPUT"
526  HTAB 5: CALL 250: PRINT "G] TURN OFF PRINTER OUTPUT"
527  HTAB 5: CALL 250: PRINT "H] DISPLAY DISK CATALOG"
528  HTAB 5: CALL 250: PRINT "I] EXIT"
534  VTAB 18: HTAB 5
535  GET A$:WH =  ASC (A$) -  ASC ("@")
536  IF WH =  - 61 THEN  HOME : END
537  IF WH < 0 THEN 535
539  ON WH GOTO 550,560,570,580,545,590,598,585,599
540  VTAB 18: HTAB 5: FLASH : CALL  - 868: PRINT "TRY AGAIN": NORMAL
     : GOTO 534
545  GOSUB 900: GOTO 505
550  PRINT D$
551  PRINT D$;"RUN DOCUMENTATION"
560  IF K2$ = "" THEN  FLASH : PRINT "NO PLAYFAIR SQUARE";: NORMAL
     : GOTO 534
562  GOSUB 600: REM  SET UP ENCIPHER
564  GOSUB 1000: REM  DO THE WORK
565  GOTO 505
570  IF K2$ = "" THEN  FLASH : PRINT "NO PLAYFAIR SQUARE";: NORMAL
     : GOTO 534
572  GOSUB 700: REM MENU FOR DECIPHER
574  GOSUB 1000: REM  DO THE WORK
575  GOTO 505
580  GOSUB 800: REM  BUILD PLAYFAIR SQUARE
581  GOTO 505
585  REM DISPLAY CATALOG
586  REM
587  HOME : PRINT D$
588  PRINT D$;"CATALOG"
589  GOSUB 1500: GOTO 505
590  REM TURN ON PRINTER AND PUT OUT A FEW BLANK LINES FOR NEATNES
     S
591 HC = 1: PRINT D$: PRINT D$;"PR#1"
592  PRINT  CHR$ (9);"72N";: PRINT : PRINT : PRINT : PRINT : PRINT

593  PRINT D$
594  PRINT D$;"PR#0"
595  GOTO 505
598 HC = 0: GOTO 505
599  HOME : END
600  REM ENCIPHER SETUP
601  REM -------------
605  HOME : VTAB 2: HTAB 5: INVERSE : PRINT "ENCIPHERING": NORMAL

606  GOSUB 2700
609  INVERSE
610  VTAB 5: HTAB 5: PRINT "CHOOSE ONE LETTER FROM EACH COLUMN": PRINT

612  NORMAL : HTAB 5: PRINT "INPUT FROM";: HTAB 25: PRINT "OUTPUT
     TO": PRINT
615  POKE 249,219: HTAB 5: CALL 250: PRINT "A] KEYBOARD";: HTAB 25
     : CALL 250
616  PRINT "C] KEYBOARD"
620  HTAB 5: CALL 250: PRINT "B] FILE";: HTAB 25: CALL 250: PRINT
     "D] FILE"
625  HTAB 25: CALL 250: PRINT "E] BOTH FILE"
626  HTAB 29: PRINT "AND"
627  HTAB 29: PRINT "KEYBOARD"
629  VTAB 15
630  HTAB 6: GET A$: PRINT A$;:W1 =  ASC (A$) -  ASC ("@")
632  IF W1 = 1 OR W1 = 2 THEN 635
633  PRINT  CHR$ (7): GOTO 629
635  VTAB 15: HTAB 26: GET B$: PRINT B$
636 W2 =  ASC (B$) -  ASC ("@")
637  IF W2 < 3 OR W2 > 5 THEN  PRINT  CHR$ (7): GOTO 635
640 IP = W1: REM  INPUT SWITCH
642 OP = W2 - 2: REM  OUTPUT SWITCH
645 PR = 1: REM  PROCESS SWITCH - INDICATES ENCIPHERMENT
```

More

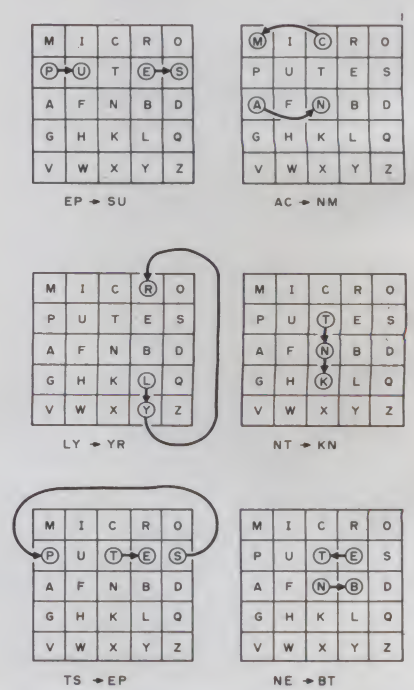


Fig. 3. Plain-text pairs to cipher-text pairs.

Listing 1 continued.

```
650  IF IP = 2 OR OP = 2 OR OP = 3 THEN  GOSUB 1600: REM  OPEN FIL
     ES
655  PRINT : HTAB 5: FLASH : PRINT "TO STOP ENCIPHERING, END"
656  HTAB 5: PRINT "INPUT LINE WITH A CONTROL-Z"
690  GOSUB 1500
699  RETURN
700  REM DECIPHER SETUP
701  REM --------------
705  HOME : VTAB 2: HTAB 5: INVERSE : PRINT "DECIPHERING": NORMAL

706  GOSUB 2700
709  INVERSE
710  VTAB 5: HTAB 5: PRINT "CHOOSE ONE LETTER FROM EACH COLUMN": PRINT

712  NORMAL : HTAB 5: PRINT "INPUT FROM";: HTAB 25: PRINT "OUTPUT
     TO": PRINT
715  POKE 249,219: HTAB 5: CALL 250: PRINT "A] KEYBOARD";: HTAB 25
     : CALL 250
716  PRINT "C] KEYBOARD"
720  HTAB 5: CALL 250: PRINT "B] FILE";: HTAB 25: CALL 250: PRINT
     "D] FILE"
725  HTAB 25: CALL 250: PRINT "E] BOTH FILE"
726  HTAB 29: PRINT "AND"
727  HTAB 29: PRINT "KEYBOARD"
729  VTAB 15
730  HTAB 6: GET A$: PRINT A$;:W1 =  ASC (A$) -  ASC ("@")
732  IF W1 = 1 OR W1 = 2 THEN 735
733  PRINT  CHR$ (7): GOTO 729
735  VTAB 15: HTAB 26: GET B$: PRINT B$
736 W2 =  ASC (B$) -  ASC ("@")
737  IF W2 < 3 OR W2 > 5 THEN  PRINT  CHR$ (7): GOTO 735
740 IP = W1: REM  INPUT SWITCH
742 OP = W2 - 2: REM  OUTPUT SWITCH
745 PR = 2: REM  PROCESS SWITCH - INDICATES DECIPHERMENT
750  IF IP = 2 OR OP = 2 OR OP = 3 THEN  GOSUB 1600: REM  OPEN FIL
     ES
755  PRINT : HTAB 5: FLASH : PRINT "TO STOP DECIPHERING, END"
756  HTAB 5: PRINT "INPUT LINE WITH A CONTROL-Z"
790  GOSUB 1500
799  RETURN
800  REM BUILD THE PLAYFAIR
801  REM  SQUARE.
802  REM ==================
805  HOME
808  FOR I = 1 TO 26:US( I ) =  - 1: NEXT I
810  INVERSE : VTAB 24: PRINT "ENTER CONTROL-Z TO RETURN IMMEDIATE
     LY";: NORMAL
815  VTAB 5: HTAB 5: PRINT "WHAT IS THE KEY?": HTAB 4: INPUT " ";K
     Y$
816  IF  LEFT$ (KY$,1) =  CHR$ (26) THEN  RETURN
817 K2$ = KY$: REM  SAVE KEY FOR LATER DISPLAY
820 LG =  LEN (KY$)
825 R = 0:C = 0
830  REM BUILD UP THE PLAYFAIR SQUARE
831  REM KEEP TRACK OF THE LETTERS
832  REM USED AS WE GO ALONG.
833  REM ==========================
834  IF LG = 0 THEN  RETURN
835  FOR L = 1 TO LG
836 C$ =  LEFT$ (KY$,1)
838  IF  LEN (KY$) > 1 THEN KY$ =  RIGHT$ (KY$, LEN (KY$) - 1)
840 X =  ASC (C$) -  ASC ("A") + 1
842  IF X < 1 OR X > 26 THEN 850
843  IF X = 10 THEN 850
845 Y = US( X )
846  IF Y = 0 THEN 850
848  GOSUB 870
850  NEXT L
855  REM NOW FILL IN THE LETTERS
856  REM WHICH DID NOT APPEAR
857  REM IN THE KEY.  BE SURE
858  REM TO IGNORE "J".
859  REM =======================
860  FOR X = 1 TO 26
861  IF X = 10 THEN 864
862 C$ =  CHR$ (X +  ASC ("A") - 1)
863  IF US( X ) < 0 THEN  GOSUB 870
864  NEXT X
865  GOSUB 890: REM  INVERT THE SQUARE
866  REM  DISPLAY THE SQUARE
867  REM ==================
868  GOSUB 900
869  RETURN
870  REM "NEXTENTRY" SUBROUTINE
871  REM ----------------------
872 L$ = "":R$ = ""
874 PF$(R,C ) = C$
876 C = C + 1: IF C > 4 THEN C = 0:R = R + 1
878 US( X ) = 0
880  RETURN
890  REM INVERT SUBROUTINE
891  REM -----------------
892  FOR I = 0 TO 4
893  FOR J = 0 TO 4
894 X =  ASC (PF$( I,J)) -  ASC ("A") + 1
895 ROW( X ) = I
896 COL( X ) = J
897  NEXT J
898  NEXT I
899  RETURN
900  REM DISPLAY THE NEWLY CREATED
901  REM PLAYFAIR SQUARE.
902  REM ==========================
905  HOME : VTAB 5: HTAB 5: INVERSE : PRINT "KEY ";: NORMAL
906  PRINT K2$
```

More

500—MAIN. The main menu of the Playfair program. This subroutine first invokes INITS to perform "re"-initializations. It then presents the user with the menu of the program's top-level options. One of these at a time is chosen and carried out by calling lower levels of the program, until the user chooses to exit (option [I]). Note that the program refuses to encipher or decipher if the user has not yet set up a Playfair square.

600—ENCIPHER SETUP. Sets up the I/O options when the user chooses to encipher.

700—DECIPHER SETUP. Sets up the I/O options when the user chooses to decipher.

800—BUILD PLAYFAIR. Constructs the Playfair square, given the user key. If the user accidentally reaches this subroutine, he can exit with "no change" to the square by entering control-Z or simply return.

The routine fills the unique letters of the key into the Playfair square, which is array PF$. It keeps track of the letters which have been used via an array—US. The entries, US(I), are initially set to −1. When a letter gets used by appearing in the key, its corresponding entry in US is set to 0. This prevents letters from being used twice. After the key is processed, a separate pass through US is made. As −1s are found in US, the program fills the letters corresponding to those entries into the remaining slots in the

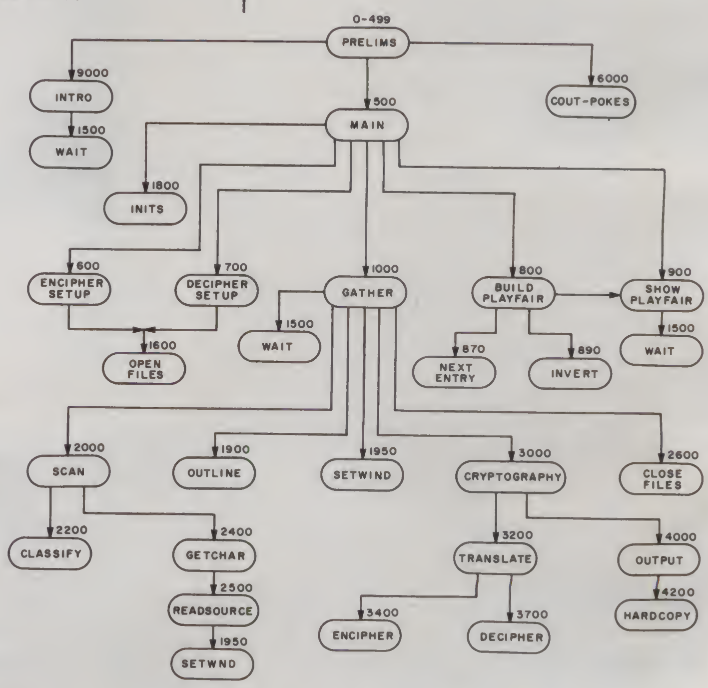


Fig. 4. Playfair program's routine-calling hierarchy.

Listing 1 continued.

```
910  VTAB 8: HTAB 5: INVERSE
911  PRINT "*************"
912  VTAB 16: HTAB 5: PRINT "*************"
915  VTAB 9
916  FOR I = 0 TO 6
917  HTAB 5: INVERSE : PRINT "*";: NORMAL : PRINT "             ";: INVERSE : P
RINT "*"
918  NEXT I
919  NORMAL
920  VTAB 10
925  FOR I = 0 TO 4
927  HTAB 7
930  FOR J = 0 TO 4
935  PRINT PF$(I,J);" ";
940  NEXT J
945  PRINT
950  NEXT I
955  VTAB 18: HTAB 5: PRINT "PLAYFAIR SQUARE"
990  GOSUB 1500
999  RETURN
1000  REM DATA GATHERING DRIVER
1001  REM RESPONSIBLE FOR BOTH
1002  REM ENCIPHERING AND
1003  REM DECIPHERING AS WELL.
1004  REM ======================
1005  GOSUB 1900: REM  OUTLINE WINDOWS
1007  REM INITIALIZE PRINTER BUFFER
1008 B$ = ""
1010  GOSUB 2000: REM  SCAN FOR FIRST CHARACTER
1015 P1$ = CH$
1020  GOSUB 2000: REM  SCAN FOR SECOND CHARACTER
1025 P2$ = CH$
1030  IF P1$ =  CHR$ (26) THEN 1090
1035  IF P2$ =  CHR$ (26) THEN P2$ = "X": GOSUB 3000: GOTO 1090
1036  REM ENSURE DISTINCT LETTERS
1037  REM IN THE DIGRAPH
1038  IF P1$ = P2$ THEN TP$ = P2$:P2$ = "X": GOSUB 3000:P1$ = TP$:
      GOTO 1020
1040  IF P1$ = "J" THEN P1$ = "I"
1041  IF P2$ = "J" THEN P2$ = "I"
1045  GOSUB 3000
1049  GOTO 1010
1090 WX = 2: GOSUB 1950
1092 T1 =  FN M6(OI):T2 =  INT (OI / 6): VTAB 12 + T2: HTAB 1 + 6 *
     T1
1094  PRINT O$;" ";:SR$ = ""
1095 LA = 1:B$ = B$ + O$: GOSUB 4200:LA = 0
1096  GOSUB 2600: REM  CLOSE FILES
1098  GOSUB 1500
1099  RETURN
1500  REM STANDARD WAIT ROUTINE
1501  REM ====================
1505  VTAB 24: FLASH : HTAB 5
1510  PRINT "PRESS ANY KEY TO CONTINUE";
1525  POKE  - 16368,0
1530  IF  PEEK ( - 16384) < 128 THEN 1530
1535  POKE  - 16368,0
1540  NORMAL
1549  RETURN
1600  REM OPEN FILES
1601  REM ==========
1605  IF IP <  > 2 THEN 1650
1610  HOME : VTAB 10: HTAB 5: INPUT "INPUT FILE? ";FI$
1615 CD$ = "OPEN " + FI$: GOSUB 1700: REM  DOS CMD TO OPEN INPUT F
     ILE
1650  IF OP <  > 2 AND OP <  > 3 THEN 1699
1655  HOME : VTAB 10: HTAB 5: INPUT "OUTPUT FILE? ";FO$
1660 CD$ = "OPEN " + FO$: GOSUB 1700: REM  DOS CMD TO OPEN OUTPUT
     FILE
1699  RETURN
1700  REM ISSUE DOS COMMANDS
1701  REM ==================
1705  PRINT  CHR$ (4);CD$
1749  RETURN
1800  REM INITIALIZE IMPORTANT
1801  REM VARIABLES.
1802  REM ===================
1805 SR$ = "": REM  SOURCE LINE-FORCES FIRST GETSOURCE
1810 CH$ = " ": REM  CURRENT INPUT CHARACTER SET TO SPACE - FORCES
      CALL TO GETCHAR
1815 OI = 0: REM  OUTPUT INDEX - I.E.
1816  REM THE POSITION OF THE CURRENT
1817  REM OUTPUT GROUP ON THE
1818  REM OUTPUT LINE.
1819  REM ========================
1820 O$ = "": REM  OUTPUT GROUP - WILL
1821  REM TO FIVE LETTERS LONG.
1822  REM =======================
1825 P1$ = " ":P2$ = " ":CH$ = " "
1830 B$ = "": REM  OUTPUT BUFFER
1831 LA = 0: REM PRINTER CONTROL SWITCH
1835 WR(1) = 2:WC(1) = 2: REM INITIAL CURSOR FOR INPUT
1836 WR(2) = 12:WC(1) = 1: REM INITIAL CURSOR FOR OUTPUT
1840  REM SET WINDOW TO FULL SCREEN
1841  REM =======================
1845  POKE 32,0: POKE 33,40: POKE 34,0: POKE 35,24
1899  RETURN
1900  REM DRAW WINDOWS TO
1901  REM SURROUND THE INPUT
1902  REM AND OUTPUT SCREENS
1903  REM ==================
1905  HOME
1906  POKE 35,10
1908  VTAB 24: HTAB 1
1910  INVERSE : PRINT "****************************************";
```

More

Listing 1 continued.

```
1917  POKE 35,24
1918  VTAB 11: HTAB 1
1920  PRINT "****************************************";
1925  VTAB 1: HTAB 1: PRINT "*************************************
 ***";
1926  POKE 34,2
1928  NORMAL : POKE 34,1: POKE 35,24
1930  VTAB 2: HTAB 1
1932  FOR I = 2 TO 10
1934  INVERSE : PRINT "*";: NORMAL : PRINT "
                    ";: INVERSE : PRINT "*";
1935  NEXT I
1936  VTAB 12: HTAB 1
1937  FOR I = 12 TO 23
1938  INVERSE : PRINT "*";: NORMAL
1939  PRINT "                                      ";: INVERSE : PRINT
 "*";
1940  NEXT I
1948  NORMAL
1949  RETURN
1950  REM SUBROUTINE TO SET
1951  REM SCROLLING WINDOWS
1952  REM =================
1955  ON WX GOTO 1970,1980
1970  REM INPUT WINDOW
1971  REM ============
1975  POKE 32,1: POKE 33,38: POKE 34,1: POKE 35,10
1979  RETURN
1980  REM OUTPUT WINDOW
1981  REM =============
1985  POKE 32,1: POKE 33,38: POKE 34,11: POKE 35,23
1999  RETURN
2000  REM SCANNER SUBROUTINE
2001  REM RETURNS CHARACTERS
2002  REM FROM THE SOURCE TEXT.
2003  REM =====================
2005  GOSUB 2400: REM  GET A CHARACTER
2010  GOSUB 2200: REM  CLASSIFY IT
2015  IF CC <  > 1 THEN 2005: REM  SKIP ALL BUT LETTERS AND CONTRO
 L-Z
2016  REM GATHERING ROUTINE CHECKS FOR ↑Z
2049  RETURN
2200  REM CLASSIFY CHARACTER
2201  REM ==================
2205  IF CH$ =  CHR$ (26) THEN CC = 1: RETURN
2206  IF CH$ =  CHR$ (13) THEN CC = 0: RETURN
2210 A =  ASC (CH$)
2215  IF A <  ASC ("A") OR A >  ASC ("Z") THEN CC = 0: RETURN
2220 CC = 1
2249  RETURN
2400  REM GET THE NEXT CHARACTER
2401  REM FROM THE SOURCE TEXT
2402  REM WE DON'T CARE WHETHER
2403  REM IT'S THE PLAIN OR THE
2404  REM CIPHER TEXT - WE JUST
2405  REM GET IT!!
2406  REM ======================
2410  IF SR$ = "" THEN  GOSUB 2500
2411  REM  NEED MORE SOURCE
2415 LG =  LEN (SR$): REM  SEE HOW MUCH LEFT
2420 CH$ =  LEFT$ (SR$,1)
2425  IF LG > 1 THEN 2435
2430  IF SR$ <  >  CHR$ (26) THEN SR$ = ""
2432  RETURN
2435 SR$ =  RIGHT$ (SR$, LEN (SR$) - 1)
2449  RETURN
2500  REM GET THE NEXT SOURCE LINE
2501  REM FROM THE CURRENTLY
2502  REM SELECTED SOURCE STREAM.
2503  REM ========================
2505 WX = 1: REM   INPUT WINDOW
2510  REM SET SCROLLING TO THE
2511  REM INPUT WINDOW
2515  GOSUB 1950
2520  IF IP = 1 THEN 2550: REM  INPUT FROM THE KEYBOARD
2522  VTAB 2: HTAB 2
2524  PRINT D$
2525  PRINT D$;"READ ";FI$
2530  INPUT SR$
2531  PRINT D$
2532  PRINT D$
2549  RETURN
2550  REM READ THE NEXT LINE FROM THE KEYBOARD
2555  HOME : VTAB 2: HTAB 2
2560  PRINT "NEXT LINE OF TEXT..."
2565  INPUT SR$
2599  RETURN
2600  REM CHECK AND CLOSE FILES
2601  REM IF NECESSARY.
2602  REM =====================
2604  PRINT D$: REM  CANCEL ANY EXISTING DOS I/O COMMAND
2605  IF IP = 2 THEN  PRINT D$;"CLOSE ";FI$
2610  IF OP = 2 OR OP = 3 THEN  PRINT D$;"WRITE ";FO$: PRINT O$; CHR$
 (26): PRINT D$;"CLOSE ";FO$
2649  RETURN
2700  REM PRINTER MESSAGE
2701  REM ===============
2705  NORMAL : VTAB 24: HTAB 5
2710  PRINT "PRINTER IS ";
2715  INVERSE
2720  IF HC = 0 THEN  PRINT "OFF";
2722  IF HC = 1 THEN  PRINT "ON";
2725  NORMAL
2749  RETURN
3000  REM CRYPTOGRAPHY
3001  REM ROUTINE.
```

More

Playfair square.

870—NEXT ENTRY. Puts a letter into the appropriate row and column of the Playfair square and advances the row and column indicators.

890—INVERT. For each letter in the Playfair square, the row and column which it occupies are recorded in the arrays ROW and COL. This provides a quick means of locating any given letter in the square without having to search the square for it.

900—SHOW PLAYFAIR. Shows the current Playfair square, and its key, on the video display.

1000—GATHER. This routine is the *boss* of the translation process. It is responsible for gathering digraphs and invoking the translation process. Its most important task is to detect the end of the input by looking for a control-Z character in the incoming character stream. In addition, it makes sure that each digraph fed to the translating routines has distinct letters, and it translates any J to an I.

1500—WAIT. Causes a pause until the user strikes a key on the Apple's terminal keyboard. It is called by many of the other routines, when it is necessary to assimilate some message on the screen. It prompts the user by flashing the message "PRESS ANY KEY TO CONTINUE" on the bottom line of the display.

1600—OPEN FILES. Uses the input and output switches (IP and OP) to detect whether input or output files are desired.

1800—INITS. Performs initializations that are necessary before each repetition of the main menu prompts.

1900—OUTLINE. The screen is subdivided into two areas, one each for input and output. This routine outlines those areas with asterisks.

1950—SETWND. This routine sets the scrolling window of the display to either the input area or the output area, whichever is appropriate. The choice is determined by the value of WX, set by the caller.

2000—SCAN. This routine calls GETCHAR and CLASSIFY until the current character is either a letter or control-Z (indicated by CC=1).

2200—CLASSIFY. This routine examines the current character and sets the variable CC to 0 or 1, depending on the class of character. Letters and control-Z are in class 1; all other characters are in class 0.

2400—GETCHAR. Picks off the next input character. Input is held in the string variable SR$. If SR$ should be exhausted (SR$=" "), then GET-

Reader Service—see page 210

Listing 1 continued.

```
3002  REM BOTH ENCIPHERING
3003  REM AND DECIPHERING
3004  REM HANDLED.
3005  REM ================
3010  GOSUB 3200: REM  TRANSLATE TO NEW DIGRAPH
3015 O$ = O$ + C1$: REM  STUFF FIRST LETTER TO OUTPUT QUEUE
3020  GOSUB 4000: REM  CHECK ON OUTPUT
3025 O$ = O$ + C2$
3030  GOSUB 4000
3049  RETURN
3099  RETURN
3200  REM TRANSLATE DIGRAPH
3201  REM BY VECTORING TO
3202  REM APPROPRIATE CIPHER
3203  REM ROUTINE BASED ON
3204  REM THE PROCESS SWITCH.
3205  REM ===================
3210  IF PR = 1 THEN  GOSUB 3400: REM  ENCIPHER
3215  IF PR = 2 THEN  GOSUB 3700: REM DECIPHER
3249  RETURN
3400  REM ENCIPHER THE CURRENT
3401  REM DIGRAPH.  NOTE:
3402  REM THE GATHERING ROUTINE
3403  REM GUARANTEES THAT P1$<>P2$
3404  REM ========================
3405 P1 =  ASC (P1$) -  ASC ("A") + 1
3410 P2 =  ASC (P2$) -  ASC ("A") + 1
3415  IF ROW(P1) <  > ROW(P2) THEN 3425
3420  GOSUB 3500: RETURN
3425  IF COL(P1) <  > COL(P2) THEN 3450
3430  GOSUB 3600: RETURN
3450  REM DIFFERENT ROW AND COLUMN
3451  REM ========================
3455 R1 = ROW(P1):R2 = ROW(P2)
3460 S1 = COL(P1):S2 = COL(P2)
3465 C1$ = PF$(R1,S2)
3470 C2$ = PF$(R2,S1)
3499  RETURN
3500  REM SAME ROW
3501  REM ========
3505 R1 = ROW(P1):R2 = ROW(P2)
3510 S1 = COL(P1) + 1: IF S1 = 5 THEN S1 = 0
3515 S2 = COL(P2) + 1: IF S2 = 5 THEN S2 = 0
3520 C1$ = PF$(R1,S1)
3525 C2$ = PF$(R2,S2)
3549  RETURN
3600  REM SAME COLUMN
3601  REM ===========
3605 S1 = COL(P1):S2 = COL(P2)
3610 R1 = ROW(P1) + 1: IF R1 = 5 THEN R1 = 0
3615 R2 = ROW(P2) + 1: IF R2 = 5 THEN R2 = 0
3620 C1$ = PF$(R1,S1)
3625 C2$ = PF$(R2,S2)
3649  RETURN
3700  REM DECIPHER THE CURRENT
3701  REM DIGRAPH
3702  REM ====================
3705 P1 =  ASC (P1$) -  ASC ("A") + 1
3710 P2 =  ASC (P2$) -  ASC ("A") + 1
3715  IF ROW(P1) <  > ROW(P2) THEN 3725
3720  GOSUB 3800: RETURN
3725  IF COL(P1) <  > COL(P2) THEN 3750
3730  GOSUB 3900: RETURN
3750  REM DIFFERENT ROW AND COLUMN
3755 R1 = ROW(P1):R2 = ROW(P2)
3760 S1 = COL(P1):S2 = COL(P2)
3765 C1$ = PF$(R1,S2)
3770 C2$ = PF$(R2,S1)
3799  RETURN
3800  REM SAME ROW
3801  REM ========
3805 R1 = ROW(P1):R2 = ROW(P2)
3810 S1 = COL(P1) - 1: IF S1 =  - 1 THEN S1 = 4
3815 S2 = COL(P2) - 1: IF S2 =  - 1 THEN S2 = 4
3820 C1$ = PF$(R1,S1)
3825 C2$ = PF$(R2,S2)
3849  RETURN
3900  REM SAME COLUMN
3901  REM ===========
3905 S1 = COL(P1):S2 = COL(P2)
3910 R1 = ROW(P1) - 1: IF R1 =  - 1 THEN R1 = 4
3915 R2 = ROW(P2) - 1: IF R2 =  - 1 THEN R2 = 4
3920 C1$ = PF$(R1,S1)
3925 C2$ = PF$(R2,S2)
3949  RETURN
4000  REM OUTPUT SUBROUTNE
4001  REM ================
4005  IF  LEN (O$) <  > 5 THEN  RETURN
4006  REM NOW SETUP OUTPUT WINDOW
4007 WX = 2: GOSUB 1950
4010 T$ = O$:O$ = "": REM  START A NEW GROUP
4012 B$ = B$ + T$ + " ": IF HC = 1 THEN  GOSUB 4200
4013  IF HC = 0 THEN B$ = ""
4015  IF OP = 2 THEN 4050
4016 T1 =  FN M6(OI):T2 =  INT (OI / 6)
4018  VTAB 12 + T2: HTAB 1 + 6 * T1
4020  PRINT T$;" ";
4025 OI = OI + 1
4026  IF OI = 72 THEN  CALL  - 912:OI = 66
4030  IF OP <  > 3 THEN  RETURN
4050  REM  DISK OUTPUT
4051  REM =============
4054  PRINT D$
4055  PRINT D$;"WRITE ";FO$
4060  PRINT T$;" ";
```

More →

CHAR invokes READSOURCE to get more.

2500—READSOURCE. Reads the next line of input from the active input stream—either the keyboard or the file selected by the user. It is important to note that if the program is fed a text file created by some means other than running the program itself, the END OF DATA message will result if the file does not end with control-Z.

2600—CLOSE FILES. Called at the end of a given encipher or decipher to close any open files.

3000—CRYPTOGRAPHY. Invokes the translation process and makes sure the results get output.

3200—TRANSLATE. Uses the process switch (PR) to invoke the appropriate translation routine.

3400—ENCIPHER. Actually carries out the encipherment of each digraph. The letters of the plain-text digraph are stored in the string variables P1$ and P2$. The resulting cipher-text letters are stored in the string variables C1$ and C2$.

3700—DECIPHER. Carries out the decipherment of each digraph. Since the scanning routines are unaware of which translation process is going on, the incoming digraph (cipher text) is still stored in the variables P1$, P2$, and the outgoing digraph (plain text) will be stored in the variables C1$, C2$.

4000—OUTPUT. This routine supervises the output. It waits until a group of five letters has been prepared, by examining the length of the string in O$. It makes sure that the output goes to just the right places, by using the switches OP and HC.

4200—HARDCOPY. Dumps output to the printer when appropriate. It waits until a buffer, B$, reaches length 72 before printing. This might have to be modified for some printers.

6000—COUT-POKES. Strictly speaking, this routine is not necessary. It pokes in a machine-language routine which enables the user to output any character desired by first poking its code into location 249. It is used by the MAIN routine and the ENCIPHER SETUP and DECIPHER SETUP routines in order to print the "[" character. This could also have been done by using the CHR$ function. A routine such as this would be necessary in Integer BASIC.

9000—INTRO. Prints a brief general introduction to the program.■

Listing 1 continued.

```
4070  PRINT D$: PRINT D$
4099  RETURN
4200  REM HARD COPY OUTPUT ROUTINE
4201  REM INVOKED WHEN THE HC FLAG
4202  REM IS TRUE (=1).
4203  REM
4205  IF HC = 0 THEN  RETURN
4207  IF LA = 1 THEN 4210
4208  IF  LEN (B$) < 72 THEN  RETURN
4210  PRINT D$
4215  PRINT D$;"PR#1"
4217  PRINT  CHR$ (9);"72N";
4220  PRINT B$;
4225  PRINT D$
4230  PRINT D$;"PR#0"
4235 B$ = ""
4249  RETURN
6000  REM POKE IN COUT
6001  REM INTERFACE ROUTINE
6002  REM =================
6005  POKE 250,165: POKE 251,249
6006  POKE 252,32: POKE 253,237: POKE 254,253
6007  POKE 255,96
6010  RETURN
6015  REM TO USE:
6016  REM
6017  REM POKE 249,CODE
6018  REM CALL 250
6109  REM
6120  REM WHERE,
6121  REM
6122  REM CODE=APPLE CHARACTER
6123  REM     CODE DESIRED
6124  REM --------------------
9000  REM  INTRODUCTION
9001  REM
9005  HOME
9010  PRINT "     THIS PROGRAM WILL DEMONSTRATE THE"
9011  PRINT "FAMOUS PLAYFAIR CIPHER. THIS IS THE"
9012  PRINT "METHOD OF ENCRYPTION USED TO SEND THE"
9013  PRINT "DISTRESS SIGNAL FROM THE SINKING PT109"
9014  PRINT "BOAT COMMANDED BY JOHN F. KENNEDY IN "
9015  PRINT "WORLD WAR II.  THE PROGRAM WILL ALLOW
9016  PRINT "YOU TO CONSTRUCT PLAYFAIR SQUARES,"
9017  PRINT "AND USE THE RESULTING SQUARES TO BOTH"
9018  PRINT "ENCODE AND DECODE SECRET MESSAGES."
9019  PRINT : PRINT "     THIS VERSION OF THE PROGRAM WILL"
9020  PRINT "ASSUME THAT YOU ARE ALREADY FAMILIAR"
9021  PRINT "WITH THE THEORY OF OPERATION OF THE"
9022  PRINT "CIPHER."
9090  GOSUB 1500
9099  RETURN
```

Listing 2.

```
100  HOME
101  INVERSE : VTAB 5
102  PRINT "FOR INFORMATION ON ONE OF THE FOLLOWING ";
105  PRINT "ROUTINES, TYPE THE CORRESPONDING LETTER:"
109  NORMAL
110  PRINT  CHR$ (219);"A] MAIN"
111  PRINT  CHR$ (219);"B] ENCIPHER SETUP"
112  PRINT  CHR$ (219);"C] DECIPHER SETUP"
113  PRINT  CHR$ (219);"D] BUILD PLAYFAIR"
114  PRINT  CHR$ (219);"E] NEXT ENTRY"
115  PRINT  CHR$ (219);"F] INVERT"
116  PRINT  CHR$ (219);"G] SHOW PLAYFAIR"
117  PRINT  CHR$ (219);"H] GATHER"
118  PRINT  CHR$ (219);"I] WAIT"
119  PRINT  CHR$ (219);"J] OPEN FILES"
120  PRINT  CHR$ (219);"K] INITS"
121  PRINT  CHR$ (219);"L] OUTLINE"
122  PRINT  CHR$ (219);"M] SETWND"
125  VTAB 8
126  HTAB 22: PRINT  CHR$ (219);"N] SCAN"
127  HTAB 22: PRINT  CHR$ (219);"O] CLASSIFY"
128  HTAB 22: PRINT  CHR$ (219);"P] GETCHAR"
129  HTAB 22: PRINT  CHR$ (219);"Q] READSOURCE"
130  HTAB 22: PRINT  CHR$ (219);"R] CLOSE FILES"
131  HTAB 22: PRINT  CHR$ (219);"S] CRYPTOGRAPHY"
132  HTAB 22: PRINT  CHR$ (219);"T] TRANSLATE"
133  HTAB 22: PRINT  CHR$ (219);"U] ENCIPHER"
134  HTAB 22: PRINT  CHR$ (219);"V] DECIPHER"
135  HTAB 22: PRINT  CHR$ (219);"W] OUTPUT"
136  HTAB 22: PRINT  CHR$ (219);"X] HARDCOPY"
137  HTAB 22: PRINT  CHR$ (219);"Y] COUT POKES
138  HTAB 22: PRINT  CHR$ (219);"Z] INTRO"
150  VTAB 24: HTAB 8
151  FLASH : PRINT "TO QUIT - HIT RETURN";
152  NORMAL
500  VTAB 22: HTAB 5
505  GET A$:WH =  ASC (A$) -  ASC ("@")
510  IF WH <  >  - 61 AND WH <  >  - 51 THEN 519
514  HOME
515  PRINT  CHR$ (4): PRINT  CHR$ (4);"RUN PLAYFAIR CIPHER"
519  IF WH < 0 THEN 525
520  ON WH GOTO 1000,1100,1200,1300,1400,1500,1600,1700,1800,1900,
     2000,2100,2200,2300,2400,2500,2600,2700,2800,2900,3000,3100,3
```

More

Listing 2 continued.

```
     200,3300,3400,3500
525  VTAB 22: HTAB 10: FLASH : PRINT "TRY AGAIN": NORMAL : GOTO 50
     0
1000  REM MAIN
1001  REM ====
1005 R$ = "MAIN":LI = 500: GOSUB 5000
1010  VTAB 5
1011  PRINT "  THIS IS THE MAIN MENU ROUTINE."
1012  PRINT "IT CALLS";: INVERSE : PRINT " INITS ";: NORMAL : PRINT
     "AND PRESENTS THE OPTIONS"
1013  PRINT "AVAILABLE TO THE USER.  THE USER SELECTS";
1014  PRINT "AN OPTION BY TYPING A LETTER.  THE"
1015  PRINT "ROUTINE CALLS LOWER LEVEL ROUTINES TO"
1016  PRINT "CARRY OUT THE FUNCTION INDICATED AND"
1017  PRINT "THEN REPEATS THE MENU."
1098  GOSUB 6000
1099  GOTO 100
1100  REM ENCIPHER SETUP
1101  REM ===============
1105 R$ = "ENCIPHER SETUP":LI = 600: GOSUB 5000
1110  VTAB 5
1111  PRINT "  SETS UP THE I/O OPTIONS FOR"
1112  PRINT "ENCIPHERING.  IT MAY CALL";
1113  INVERSE : PRINT " OPEN FILES ";: NORMAL : PRINT "IF INDICATE
     D."
1198  GOSUB 6000
1199  GOTO 100
1200  REM DECIPHER SETUP
1201  REM ==============
1205 R$ = "DECIPHER SETUP":LI = 700: GOSUB 5000
1210  VTAB 5
1211  PRINT "  SETS UP THE I/O OPTIONS FOR"
1212  PRINT "DECIPHERING.  IT MAY CALL";
1213  INVERSE : PRINT " OPEN FILES";: NORMAL : PRINT " IF INDICATE
     D."
1298  GOSUB 6000
1299  GOTO 100
1300  REM BUILD PLAYFAIR
1301  REM ==============
1305 R$ = "BUILD PLAYFAIR":LI = 800: GOSUB 5000
1310  VTAB 5
1311  PRINT "  BUILDS A NEW PLAYFAIR SQUARE AFTER"
1312  PRINT "PROMPTING THE USER FOR A KEY."
1313  PRINT "CALLS";: INVERSE : PRINT " NEXT ENTRY ";
1314  NORMAL : PRINT "AND";: INVERSE : PRINT " INVERT.": NORMAL
1398  GOSUB 6000
1399  GOTO 100
1400  REM NEXT ENTRY
1401  REM ==========
1405 R$ = "NEXT ENTRY":LI = 870: GOSUB 5000
1410  VTAB 5
1411  PRINT "  ENTERS A LETTER INTO THE PLAYFAIR"
1412  PRINT "SQUARE AND ADVANCES THE ROW AND COLUMN"
1413  PRINT "INDICATORS."
1498  GOSUB 6000
1499  GOTO 100
1500  REM INVERT
1501  REM ======
1505 R$ = "INVERT":LI = 890: GOSUB 5000
1510  VTAB 5
1511  PRINT "  'INVERTS' THE PLAYFAIR SQUARE BY "
1512  PRINT "STORING THE ROW AND COLUMN LOCATIONS"
1513  PRINT "OF EACH LETTER IN SEPARATE ARRAYS. THIS"
1514  PRINT "ALLOWS QUICK LOOKUP OF A GIVEN LETTER"
1515  PRINT "IN THE SQUARE."
1598  GOSUB 6000
1599  GOTO 100
1600  REM SHOW PLAYFAIR
1601  REM =============
1605 R$ = "SHOW PLAYFAIR":LI = 900: GOSUB 5000
1610  VTAB 5
1611  PRINT "  DISPLAYS THE CURRENT PLAYFAIR SQUARE"
1612  PRINT "AND ITS KEY."
1698  GOSUB 6000
1699  GOTO 100
1700  REM GATHER
1701  REM ======
1705 R$ = "GATHER":LI = 1000: GOSUB 5000
1710  VTAB 5
1711  PRINT "  THE 'BOSS' OF THE TRANSLATION PROCESS.";
1712  PRINT "IT GATHERS DIGRAPHS AND FEEDS THEM TO "
1713  PRINT "LOWER LEVEL TRANSLATION ROUTINES."
1714  PRINT "  IT GUARANTEES THE UNIQUENESS OF THE"
1715  PRINT "LETTERS IN EACH DIGRAPH AND DETECTS THE"
1716  PRINT "END OF THE INPUT.  IT CALLS";
1717  INVERSE : PRINT " WAIT";: NORMAL
1718  PRINT ",";: INVERSE : PRINT " SCAN";: NORMAL
1719  PRINT ",": INVERSE : PRINT " OUTLINE";: NORMAL
1720  PRINT ",";: INVERSE : PRINT " SETWND";: NORMAL
1721  PRINT ",";: INVERSE : PRINT " CRYPTOGRAPHY";: NORMAL
1722  PRINT ", AND": INVERSE : PRINT " CLOSE FILES.": NORMAL
1798  GOSUB 6000
1799  GOTO 100
1800  REM WAIT
1801  REM ====
1805 R$ = "WAIT":LI = 1500: GOSUB 5000
1810  VTAB 5
1811  PRINT "  PAUSES UNTIL THE USER STRIKES A KEY"
1812  PRINT "ON THE APPLE ]"; CHR$ (219);" KEYBOARD."
1898  GOSUB 6000
1899  GOTO 100
1900  REM OPEN FILES
1901  REM ==========
1905 R$ = "OPEN FILES":LI = 1600: GOSUB 5000
1910  VTAB 5
```

More

✓Reader Service—see page 210

Listing 2 continued.

```
1911  PRINT "  OPENS EITHER A SINGLE INPUT FILE OR A"
1912  PRINT "SINGLE OUTPUT FILE, OR BOTH.  IT USES"
1913  PRINT "THE VALUES OF THE SWITCHES IP AND OP"
1914  PRINT "FOR GUIDANCE.  THESE ARE SET BY THE"
1915  PRINT "ROUTINES";: INVERSE : PRINT " ENCIPHER SETUP ";: NORMAL
   : PRINT "AND"
1916  INVERSE : PRINT " DECIPHER SETUP.": NORMAL
1998  GOSUB 6000
1999  GOTO 100
2000  REM INITS
2001  REM =====
2005 R$ = "INITS":LI = 1800: GOSUB 5000
2010  VTAB 5
2011  PRINT "  PERFORMS RE-INITIALIZATION BEFORE "
2012  PRINT "EACH SELECTION OF A MENU OPTION.  MOST"
2013  PRINT "IMPORTANT ARE VARIABLES DEALING WITH "
2014  PRINT "THE TRANSLATION PROCESS, SUCH AS SR$,"
2015  PRINT "THE INPUT TEXT LINE; AND SWITCHES FOR"
2016  PRINT "SUCH THINGS AS I/O.  NOTE: THE PRINTER"
2017  PRINT "(HARDCOPY) SWITCH IS NOT RE-INITIALIZED"
2018  PRINT "EXCEPT BY THE USER."
2098  GOSUB 6000
2099  GOTO 100
2100  REM OUTLINE
2101  REM =======
2105 R$ = "OUTLINE":LI = 1900: GOSUB 5000
2110  VTAB 5
2111  PRINT "  OUTLINES THE SCREEN WITH ASTERISKS,"
2112  PRINT "SUBDIVIDING IT INTO TWO 'WINDOWS', ONE"
2113  PRINT "FOR INPUT TEXT AND ONE FOR OUTPUT TEXT."
2198  GOSUB 6000
2199  GOTO 100
2200  REM SETWND
2201  REM ======
2205 R$ = "SETWND":LI = 1950: GOSUB 5000
2210  VTAB 5
2211  PRINT "  SETS THE APPLE'S SCROLLING WINDOW TO"
2212  PRINT "EITHER THE INPUT OR THE OUTPUT PORTION"
2213  PRINT "OF THE SCREEN."
2298  GOSUB 6000
2299  GOTO 100
2300  REM SCAN
2301  REM ====
2305 R$ = "SCAN":LI = 2000: GOSUB 5000
2310  VTAB 5
2311  PRINT "  REPEATEDLY CALLS";: INVERSE : PRINT " GETCHAR ";: NORMAL
   : PRINT "AND"
2312  INVERSE : PRINT "CLASSIFY ";: NORMAL : PRINT "UNTIL THE CURR
   ENT CHARACTER"
2313  PRINT "IS EITHER CONTROL-Z OR A LETTER.  THIS "
2314  PRINT "IS INDICATED BY THE VALUE OF THE"
2315  PRINT "VARIABLE CC. IF CC=1, THEN SCAN HAS DONE";
2316  PRINT "ITS JOB."
2398  GOSUB 6000
2399  GOTO 100
2400  REM CLASSIFY
2401  REM ========
2405 R$ = "CLASSIFY":LI = 2200: GOSUB 5000
2410  VTAB 5
2411  PRINT "  SETS THE VARIABLE CC (ABBREVIATION"
2412  PRINT "FOR 'CHARACTER CLASS') TO EITHER 0 OR 1"
2413  PRINT "DEPENDING ON THE CLASS OF THAT CHARACTER";
2414  PRINT "LETTERS AND CONTROL-Z ARE IN CLASS 1."
2415  PRINT "ALL OTHER CHARACTERS ARE IN CLASS 0."
2498  GOSUB 6000
2499  GOTO 100
2500  REM GETCHAR
2501  REM =======
2505 R$ = "GETCHAR":LI = 2400: GOSUB 5000
2510  VTAB 5
2511  PRINT "  GETS THE NEXT CHARACTER FROM SR$, THE"
2512  PRINT "INPUT LINE, IF THERE IS ONE.  IF NOT"
2513  PRINT "(SR$=''), IT CALLS";: INVERSE : PRINT " READSOURCE ";
   : NORMAL : PRINT " TO"
2514  PRINT "RETURN THE NEXT LINE OF TEXT."
2598  GOSUB 6000
2599  GOTO 100
2600  REM READSOURCE
2601  REM ==========
2605 R$ = "READSOURCE":LI = 2500: GOSUB 5000
2610  VTAB 5
2611  PRINT "  READS THE NEXT LINE OF TEXT FROM THE"
2612  PRINT "CURRENTLY ACTIVE INPUT MEDIUM - EITHER"
2613  PRINT "KEYBOARD OR FILE.  IT USES THE IP SWITCH";
2614  PRINT "TO GUIDE ITSELF."
2698  GOSUB 6000
2699  GOTO 100
2700  REM CLOSE FILES
2701  REM ===========
2705 R$ = "CLOSE FILES":LI = 2600: GOSUB 5000
2710  VTAB 5
2711  PRINT "  CLOSES ANY OPEN FILES AT THE END OF"
2712  PRINT "EACH INDIVIDUAL TRANSLATION SESSION."
2713  PRINT "THUS, IT MAY BE INVOKED MANY TIMES "
2714  PRINT "DURING THE SAME PROGRAM RUN."
2798  GOSUB 6000
2799  GOTO 100
2800  REM CRYPTOGRAPHY
2801  REM ============
2805 R$ = "CRYPTOGRAPHY":LI = 3000: GOSUB 5000
2810  VTAB 5
2811  PRINT "  INVOKES THE TRANSLATION PROCESS AND"
2812  PRINT "MAKES SURE THE RESULTS ARE OUTPUT."
2898  GOSUB 6000
2899  GOTO 100
```

More

Listing 2 continued.

```
2900  REM TRANSLATE
2901  REM =========
2905 R$ = "TRANSLATE":LI = 3200: GOSUB 5000
2910  VTAB 5
2911  PRINT "  USES THE PROCESS SWITCH (PR) TO DETECT";
2912  PRINT "WHICH TRANSLATION ROUTINE TO INVOKE."
2913  PRINT "CALLS BOTH";: INVERSE : PRINT " ENCIPHER ";: NORMAL :
      PRINT "AND";: INVERSE : PRINT " DECIPHER.": NORMAL
2998  GOSUB 6000
2999  GOTO 100
3000  REM ENCIPHER
3001  REM ========
3005 R$ = "ENCIPHER":LI = 3400: GOSUB 5000
3010  VTAB 5
3011  PRINT "  CARRIES OUT THE DETAILS OF CONVERTING"
3012  PRINT "PLAIN TEXT PAIRS TO CIPHER TEXT PAIRS."
3098  GOSUB 6000
3099  GOTO 100
3100  REM DECIPHER
3101  REM ========
3105 R$ = "DECIPHER":LI = 3700: GOSUB 5000
3110  VTAB 5
3111  PRINT "  CARRIES OUT THE DETAILS OF CONVERTING"
3112  PRINT "CIPHER TEXT PAIRS TO PLAIN TEXT PAIRS."
3198  GOSUB 6000
3199  GOTO 100
3200  REM OUTPUT
3201  REM ======
3205 R$ = "OUTPUT":LI = 4000: GOSUB 5000
3210  VTAB 5
3211  PRINT "  SUPERVISES OUTPUT TO THE VARIOUS"
3212  PRINT "DEVICES.  IT WAITS UNTIL 5 LETTERS HAVE"
3213  PRINT "BEEN ACCUMULATED.  THIS IS TO ENSURE "
3214  PRINT "THE 'STANDARD' CRYPTOGRAPHIC OUTPUT"
3215  PRINT "FORMAT."
3216  PRINT "  CALLS";: INVERSE : PRINT " HARDCOPY.": NORMAL
3298  GOSUB 6000
3299  GOTO 100
3300  REM HARDCOPY
3301  REM ========
3305 R$ = "HARDCOPY":LI = 4200: GOSUB 5000
3310  VTAB 5
3311  PRINT "  PRODUCES THE PRINTER OUTPUT.  IT WAITS";
3312  PRINT "UNTIL A BUFFER (B$) FILLS WITH 72 "
3313  PRINT "CHARACTERS BEFORE SENDING IT TO THE"
3314  PRINT "PRINTER."
3398  GOSUB 6000
3399  GOTO 100
3400  REM COUT POKES
3401  REM ==========
3405 R$ = "COUT POKES":LI = 6000: GOSUB 5000
3410  VTAB 5
3411  PRINT "  POKES IN AN INTERFACE ROUTINE TO THE"
3412  PRINT "MONITOR COUT SUBROUTINE."
3498  GOSUB 6000
3499  GOTO 100
3500  REM INTRO
3501  REM =====
3505 R$ = "INTRO":LI = 9000: GOSUB 5000
3510  VTAB 5
3511  PRINT "  INTRODUCES THE PROGRAM."
3598  GOSUB 6000
3599  GOTO 100
5000  REM SETUP
5001  REM =====
5005  HOME : INVERSE
5010  VTAB 2: PRINT "ROUTINE NAME:"
5015  PRINT "STARTS AT:    "
5020  NORMAL
5025  VTAB 2: HTAB 19: PRINT R$
5030  HTAB 19: PRINT LI
5049  RETURN
6000  REM STANDARD WAIT ROUTINE
6001  REM =====================
6005  VTAB 24: FLASH : HTAB 5
6010  PRINT "PRESS ANY KEY TO CONTINUE";
6015  POKE  - 16368,0
6020  IF  PEEK ( - 16384) < 128 THEN 6020
6025  POKE  - 16368,0
6030  NORMAL
6049  RETURN
```

Challenge Cipher

For those readers who enjoy cryptanalysis, I present in closing the following cipher text, created using the Playfair program. If you give up, turn to page 192 to find the key.

RAPYN AEPGT THOQC GOKXS BOXSP TMAOC OQCGG
DYCPM MGOYI NAUAI LDGDA EGVGM DOAQI KAUGV
DLTCB QMAMP BASID RTUDO RNWOB ASITE FOKBD
OFIMG ELOQC PCTLR GTUTF AGMIP IYKDL TSILC
DCMAV ANISI DGDRS IXSDR GRLRL CFODC ABQUC
GUAGV AMKGB TPSSI GMIPQ IKHYA CDAMB TSYXX

FINE TOOLS FOR YOUR TRS-80*

ULTRA-MON

ULTRA-MON is a unique and powerful machine-language monitor. It is ROM independent and will function in Level II or DOS BASIC. With ULTRA-MON, you will be able to write, modify, study and debug machine-language programs. Plus, you'll be able to avoid the frustrations and "bomb-outs" usually associated with machine-language programming. ULTRA-MON displays, disassembles, traces (hardcopy trace disassembly, too!), modifies, relocates memory, prints and even relocates itself with simple commands. Using interpretive execution, ULTRA-MON allows you to put breakpoints in ROM. This powerful monitor can even fetch, decode, disassemble and analyze each instruction individually so that your program cannot bomb out. ULTRA-MON is designed for the beginning machine-language programmer as well as the professional. The documentation contains a Simple Demonstration section geared to the novice programmer. Consequently, the program is a learning device as well as an extremely useful programming tool.
If you are serious about programming, you need to add this powerful utility to your library today. (T1)

Order No. 5003R $24.95.

UTILITY I

Here's software utility that can make programming easier for you. These powerful tools will take much of the frustration out of essential operations. You'll wonder how you managed without them. They'll stamp your work with the mark of professionalism.
RENUM—Now you can easily renumber any Level II program to make room for modification or to clean up the listing. (Only for use with 16K of memory.)
DUPLIK—This program will let you duplicate any BASIC, assembler, or machine-language program, verify the data, and even copy Level I programs on a Level II machine.

See how efficient and well-ordered your programming becomes. (T1)

Order No. 0081R $9.95.

UTILITY II

One of the most popular and useful of the utility packages, Utility II is the second in a series of programs designed to take the drudgery out of editing.

This combination of programs adds flexiblity to your BASIC programs by allowing you to combine them with other BASIC programs, or with machine-language programs or routines.
CFETCH—You'll be able to merge BASIC programs, with consecutive line numbers, into one program. CFETCH can also search through any Level II program tape and display the file name for all the programs.
CWRITE—Combine subroutines that work in different memory locations into one program. CWRITE works with BASIC and/or one or more machine-language programs. It will even give you a general checksum to verify that your program hasn't dropped any bits.

Use these two programs as your dependable "assistants" to speed merging processes and as a totally reliable retrieval means in search and edit operations.

A fine tool for your TRS-80! (T1)

Order No. 0076R $9.95.

TLDIS & DLDIS

You've bought a super machine-code program, but now wonder how it works. Maybe you even used a quick PEEK routine to glance through it when it was in memory. If so, you definitely noticed the complete lack of comments in the code, making it almost impossible for you to decipher and understand it.

Well, Instant Software's Labeling Disassemblers are the answer to your problem.

TLDIS (Tape-based Labeling DISassembler) and DLDIS (Disk-based Labeling DISassembler) are three-pass, label-assigning disassemblers which assign labels (where appropriate) to the routines in a machine-language program. Their output is almost identical to that of a hand-assembled source code.

You can send the disassembly to a line-printer (Radio Shack parallel port) for either TLDIS or DLDIS. (The difference between these utilities is the storage mode of the disassembly.)

TLDIS can send the disassembly to cassette tape, DLDIS can send it to disk; both send it to the video monitor. The stored disassembly from TLDIS may be reassembled with Radio Shack's EDTASM™—the disassembly from DLDIS, with Apparat's extension of EDTASM™. Because of the use of labels, it is a simple matter to change any object code program by disassembling it and then making changes to the resultant source code, without losing track of jump/load addresses. Labels start with "AA00" and increment up, in even numbered steps (AA02, AA04, etc.). The odd numbers (AA01, AA03, etc.) are left for you to use for the source code during reassembly.

The printing of the disassembly may be temporarily halted by using [SHIFT] @ (just as in BASIC) or it may be ended by pressing the [BREAK] key. It also has a comments column to display ASCII characters used in a LD or CP opcode.

TLDIS and DLDIS may be relocated in memory to avoid conflict with the program you disassembler.

The next time you need to "climb inside" a machine-code program, take DLDIS or TLDIS with you. We promise that it will be an easier journey.

Order No. 0230R (TLDIS) $14.95 (T1)
Order No. 0231RD (DLDIS) $19.95 (T2)

COMPRESSION UTILITY PACK

Do you want to add sound routines to a Space Trek program that already uses 16K? Or maybe you need an extra column in that financial report program, but when you run it, you get OM errors?

With a wave of your hand, and a little help from either of the COMPRESSION programs in this package, your problems are over.

COMPRESS-80—fits in 265 bytes, deletes spaces, and offers the choice of leaving REMark line numbers in the program, or deleting them altogether.

SUPERCOMPRESS—uses 767 bytes and can do everything COMPRESS-80 can do PLUS, it packs the program into the smallest possible number of multiple statement lines.

With the Compression Utility Pack and your own programming skill, you can add all those little extras to your BASIC programs. (T1)

Order No. 0246R $19.95.

TO ORDER:

See Your Local
Instant Software Dealer
or
Order Directly:
CALL TOLL FREE
1-800-258-5473

*A trademark of Tandy Corporation

Instant Software™ ✓40

PETERBOROUGH, N.H. 03458
603-924-7296

FOR OSI COMPUTERS

★ SMARTERM ★

An 'Intelligent' Terminal Software System for OSI Computers by Phil Lindquist

- Even, Odd, or No Parity Use-Selected
- Full or Half Duplex Operation
- User Controlled Storage of Incoming Data
- Stored Data Printed or Saved on Disk
- Saved Data Recalled For Review or Output
- Control Key Initiation of LOGON Messages

- Join The Micro Communications Revolution -

C2-4P MF, C4P MF Systems $19.95
C2-8P DF, C8P DF, C4P DF Systems $23.95

Order From:

COMPUTER CONNECTION, INC.
38437 Grand River • Farmington Hills, Mich. 48018

✓278

Dealer Inquiries Welcome

The Joy of Computing Series

Dedicated to personal computing.

Coming in June:

Real Time Programming: Neglected Topics

Caxton C. Foster, University of Massachusetts, Amherst

Real Time Programming is an original approach to mastering the programming skills needed to connect microcomputers to other computers and access their programs. Attention focuses on interrupts, input-output ports, digital filters, servomechanisms, and communication over limited pathways. 1981 (01937) c.224 pp. Paper $8.95 (tent.)

New in April:

Executive Computing: How to Get It Done on Your Own

John M. Nevison, John M. Nevison Associates
1981 (05248) 336 pp. Paper $8.95

Computer Consciousness: Surviving the Automated 80s

H. Dominic Covvey and Dr. Neil H. McAlister, Toronto General Hospital
1980 (01939) 192 pp. $5.95

A Bit of BASIC

Thomas A. Dwyer and Margot Critchfield, University of Pittsburgh
1980 (03115) 184 pp. $5.95

We have sold more than 75,000 copies of the three original books in the JOY series:

BASIC and the Personal Computer

Thomas A. Dwyer and Margot Critchfield, University of Pittsburgh
1978 (01589) 350 pp. Paper $12.95

Programming a Microcomputer: 6502

Caxton C. Foster, University of Massachusetts, Amherst
1978 (01995) 256 pp. Paper $8.95

The Little Book of BASIC Style: How to Write a Program You Can Read

John M. Nevison, John M. Nevison Associates
1978 (05247) 160 pp. Paper $5.95

Available at your bookstore or computer store. Or write to Thomas A. Bell, Marketing Manager, for **10-day free trial examination.**

Business & Professional Division
Addison-Wesley Publishing Company, Inc. ✓2
Reading, Massachusetts 01867

NEW PRICES...
NEW SOFTWARE for

OHIO SCIENTIFIC

(Prices effective Dec. 1, 1980)

WP6502 Word Processor Version 1.2

- Cassette (all systems) .... $ 50
- 5 Inch diskette (all systems) $100
- 8 Inch disk for OS65D Systems $100
- 8 Inch disk for OS65U Systems $100
- Programmed Instruction Training Course...... $ 25

Version 1.2 Enhancements

- DQ Mail (DMS Interface) .... $ 75
- DQ Duplicator (5" single drive copier) .... $ 30
- DQ Duplicator for 8" .... $ 30
- DQ Justify (TTY & Diablo Logic)
  Diablo/Qume & all NEC Spinwriters
  65D.................................. $ 50
  65U.................................. $ 50
- DQ Justify (TTY & Proportional Spacing)
  Parallel NEC Spinwriters & Centronics 737
  65D.................................. $ 50
  65U.................................. $ 50
- DQ Secretary. Dwo's enhanced 65D.......... $ 50
- DQ Secretary. Dwo's enhanced 65U for single user, timesharing and networking systems $250

WP6502 Word Processor Version 1.3
(January 1, 1980)

- 65D $250
- 65U $250
- Multi-User & Networking $500
- Programmed Instruction Training Course...... $ 25

Descriptive Brochure for all products.........FREE

Dwo Quong Fok Lok Sow
23 East 20th Street
New York City, N.Y. 10003 ✓87
(212) 673-6310

This hardware modification for the Challenger 1P or Superboard II quickly and inexpensively readies your system for a printer or modem.

Energize Those OSI Peripheral Ports

By Edward C. Jones

If you own an OSI Challenger 1P or Superboard II, you might wonder about those Molex connectors on the circuit board and whether you can hook up a printer or modem to them. You might have actually tried to hook up a printer to the connectors, only to discover that it will not work. Take heart: you can have those ports working in about two hours and with $10 to $20 in parts. Use the plastic Molex connectors, J2 and J3, on the circuit board to provide these signals. The mod will activate both ports, and provide switch-selectable user control of incoming data.

Preparation

Standard CIP/Superboards have the holes already drilled to accept components, but do not come equipped with them. See the parts list for the required hardware.

For C1P owners (Superboard owners can skip this paragraph), the first step is to remove the circuit board. Disconnect the cassette cables, the video cable and the power cord. Turn the computer over and remove the six screws. Remove the cover. Remove the six screws that hold the circuit board to the case. Take the board out of the case and put it on your work surface.

Modifications

Look at Fig. 1. First, install the additional components. Position your circuit board so that the keyboard is facing away from you. Looking at the rear of the board, locate the J2 connector, which is the connector that your cassette recorder plugs into. Just to the left of the connector are a lot of empty holes. This is the area that you'll be working in.

First, locate the correct area shown for the 12 resistors in two rows, R38 through R49 on the OSI schematic. Obtain all of the 220 ohm and 390 ohm resistors, and bend one of the leads as shown. Insert one each of the 220 ohm resistors into the two holes at locations R38, R40, R42, R44, R46 and R48.

Next, insert one each of the 390 ohm resistors into the two holes at locations R39, R41, R43, R45, R47 and R49.

Now insert the two 14-pin sockets into the holes at locations U67 and U68, just above the new resistors. It does not matter which way you install the sockets.

Now lightly tape the resistors and sockets to the circuit board. Then turn the board upside down and sol-

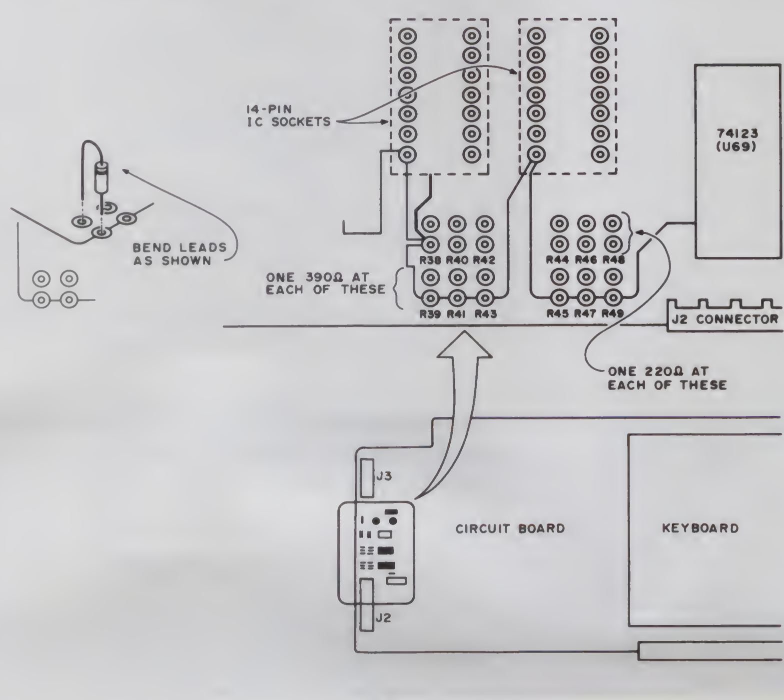


Fig. 1.

Edward C. Jones, 2334 Antiqua Court, Reston, VA 22091.

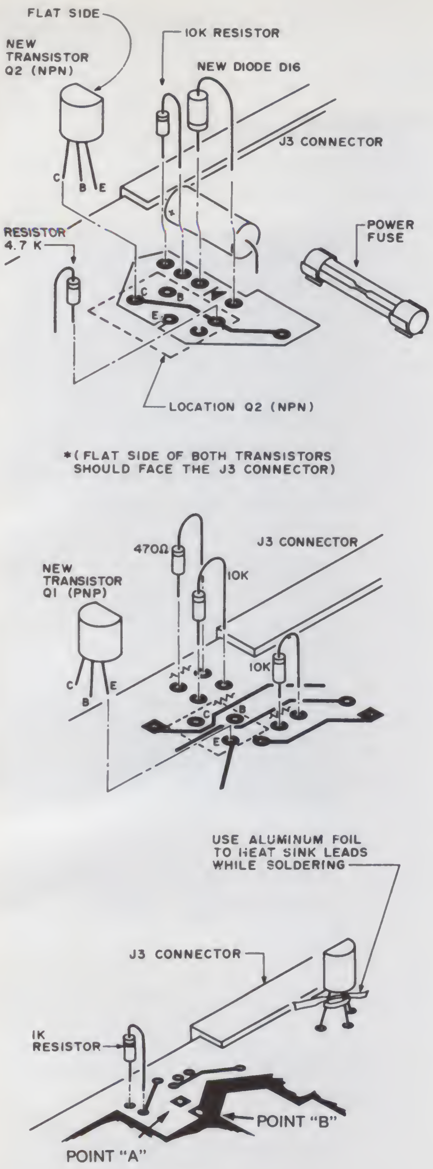


Fig. 2.

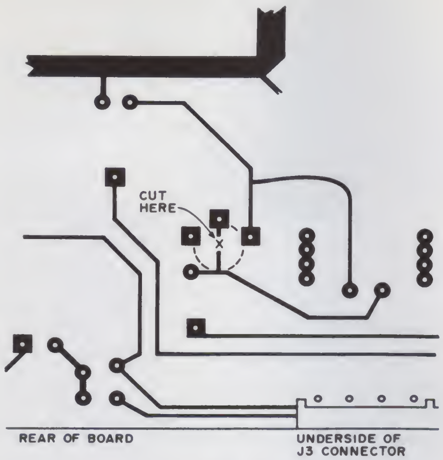


Fig. 3.

der the new components to the connection points. Cut the excess leads from the resistors, turn the board back over and remove the tape.

Look at Fig. 2 and locate the holes for the transistors at locations Q1 and Q2, and for the diode at location D16. Note the layout of the transistor leads and the diode leads. (Be careful not to mix up the npn and the pnp transistors.) Looking at the circuit board from the rear, orient the transistors as shown, and insert the pnp transistor into the Q1 location, and the npn transistor into the Q2 location. Insert the diode into the two holes at location D16. The end with the band should be facing the rear of the circuit board.

You should have six resistors remaining—470 ohm, 1k, 4.7k and three 10k. Bend the leads in the same manner as the other resistors and insert them in the holes as shown in the figure. Check that the transistors and diode are inserted correctly. Cut two strips of aluminum foil, approximately ¼ by 3 or 4 inches. Twist and wrap the strips around the transistor leads as shown. Lightly tape the transistors, resistors and the diode to the board; then turn the board upside down and solder the leads to the connections. Cut the excess leads. Leave the board upside down.

Next come the switch wiring and mounting. Locate the area shown in Fig. 3. You'll need to cut the wire shown connecting these two points. *Carefully* use a small knife or file, and cut the wire run shown on the circuit board.

Now turn the board right side up, remove the aluminum foil from the transistors, and locate the wiring area just to left of the #1 key on the keyboard. Cut the wire run as shown in Fig. 4. Connect two wires, about 12 inches in length, to the two terminals of the SPST switch. Solder the other ends of the two wires to the connection holes on the board, as shown in the figure.

Now, turn the board so that the rear is again facing you. Locate the fuse near connector J3. In front of the fuse is a capacitor. Look to the immediate right of the capacitor, and you will see three holes as shown in Fig. 5. Connect one end of a 12-inch piece of wire to each of the three poles of the rotary switch, and one to the common, as shown. Solder the wires from each of the poles of the switch to the board as shown.

Now, look at pin 4 of connector J3 (Fig. 6). Follow the pin out of the back of the connector. Locate the wire run and connection point tied to pin 4, as shown in the figure. Solder the common lead from the switch to this connection point. Next, insert the 7414 IC in the left-hand socket, and the 74LS17 IC in the right-hand socket. Make sure that the notch, or dot, on one end of the ICs faces the keyboard side of the circuit board.

All you need to do now is mount the switches somewhere and connect the peripheral of your choice to the port of your choice. Mounting the switches is easy if you have a Superboard II. Depending on how your case is designed, it's up to you. On a C1P, you can drill two holes at the rear of the cabinet in an area that has sufficient space, and mount the switches. You will also need to provide another small hole for the plug that matches your printer, modem or whatever.

Conclusion

At this point, I am about to leave you on your own, but let me explain what we have done so that you're not totally in the dark. When the rotary switch is in the pole 1 position, your system loads from cassette, as it did previously. In the pole 2 position, the RS 232/modem, or J3 connector, is

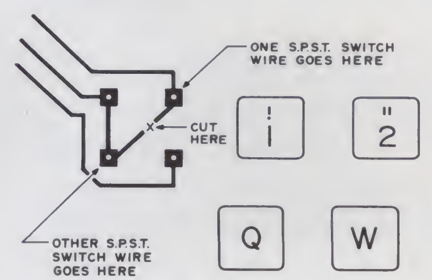


Fig. 4.

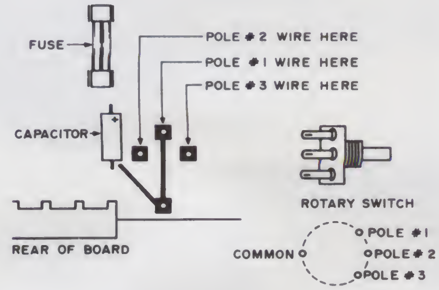


Fig. 5.

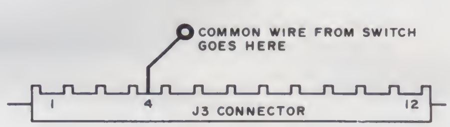


Fig. 6.

switched in. In the pole 3 position, the printer port, or J2 connector, is switched in. The SPST switch, when closed, will enable the CTS (clear to send) function. The rotary switch will control the receive data, so that information could not come into the computer from two sources at the same time. The transmit data will always be present at both J2 and J3 any time you execute a save command.

So, if a printer is hooked up to the J2 port and you type in SAVE and turn on the printer, the data that appears on screen will print on the printer. The clear-to-send switch should normally be left in the closed position. In fact, it *must* be in the closed position when using your cassette deck to load programs. It is only necessary to turn the switch to the open position when using a printer or modem that requires the use of a clear-to-send signal.

The big question that may now remain is, "but how do I hook up my printer?" I wish I could answer that. But, since every peripheral out there is different, there is no one answer to that question. The information in Table 1 will hopefully help you figure out how to do this after comparing the outputs with the inputs needed by your peripheral.

Most printers will need two or three lines; a ground line, a transmit data line or RS-232 data out line; and possibly a clear-to-send line. On some printers, this CLS line is called BUSY or ACKNOWLEDGE.

Most telephone modems will need both the RS-232 lines, along with clear-to-send and request-to-send, depending on the modem's duplex setup. If the peripheral is other than a modem or printer, get a full set of spec sheets, because you're on your own.

The RS-232 output lines conform to a standard positive voltage level. Unfortunately, a few printers require a negative voltage level. If your printer is one of them, you can use the RS-232 port to provide an output line to the printer, provided you do two more tasks. Locate points A and B in Fig. 2 for mounting the 1k, 4.7k and 10k resistors. On the underside of the board is a wire run between these two points. You will have to cut it. Then, connect a negative voltage supply between this point and ground, by wiring in a 9-volt battery. The positive lead goes to pin 8, and the negative lead goes to pin 7 of J2. Add a switch to stop battery drain when you're not using the printer.

Also, keep in mind that these are serial ports. They are designed to work with any peripherals that accept serial data, which is most of them. There are some peripherals that require parallel inputs. *Stay* away from them, if at all possible. You would need considerably more circuitry than is detailed here to make a parallel peripheral work with your serial outputs. If you are into hardware kludging, you can try an interface using a UART chip. ■

Parts List

- 1 7417 IC
- 1 74LS14 IC
- 6 220 ohm resistors
- 6 390 ohm resistors
- 1 470 ohm resistors
- 1 1k ohm resistor
- 1 4.7k ohm resistor
- 3 10k ohm resistors
- 2 14-pin IC DIP sockets
- 1 1N914 diode
- 1 2N2222 transistor (Radio Shack 276-2009)
- 1 2N2907 transistor (Radio Shack 276-2023)
- 1 12-pin Molex connector (like the one that connects your cassette recorder to your circuit board)
- 1 single pole, three-throw rotary switch
- 1 SPST switch
- 1 connector to match your printer, modem or whatever.

miscellaneous hardware: hookup wire (no. 22 or 20 gauge), solder, soldering iron, electrical tape, a small piece of aluminum foil.

Connectors (J2)

Pin #	
1	Rec data (output to printer or peripheral)
2	Rec clock
3	Clear to send
4	Transmit data (input from peripheral)
5	Transmit clock
6	Request to send

7 through 12—Cassette and video connections. Pins 8 and 11 are ground. A ground line will need to go to the peripheral.

Connectors (J3)

1	Ground
2	RS-232 data output
3	RS-232 data input
4	Rec data
5	Rec data #1
6	Rec data #2
7	Negative voltage level for RS-232 (see text)
8	not used
9	Clear to send
10	Clear to send #2
11	not used
12	not used

Table 1.


Whatever happened to eenie, meenie, miney, mo?
This may put the Godfather out of business.
I could be another Solomon...
If only my heart would stop racing...
It must use Bayesian, weighted factor analysis, and...
Brilliant! Like a window into the future.
..a perfect ift for that rban cowgirl!
laybe this'll elp me choose career...
could se it to elect my staff.
Would I rather have Winston's millions or Billy Joe's love?
Hmmmm... could be my ticket to the Boardroom.
Can't any of these people afford $29.95?
Should I buy stock or commodities in this economy?
The decision is...

Here's a way to get into the rat race with a new perspective on Apple II's high-resolution graphics.

A Rat's-Eye View of Mazes

By Brian McCarson

Behavioral scientists often place rats into mazes to test their reactions. But except at a few elegant British gardens, most human beings never see a maze from the inside. Now, thanks to the Renaissance invention of perspective drawing and the more recent development of microcomputer graphics displays, science can offer man the same perplexing experience as the rat.

Rats is a program for the Apple II that renders a rat's-eye view of the interior of a maze, using the Apple's high-resolution graphics capability. The user is forced to find his way through the twists and turns to the exit without benefit of other orientation.

Program Use

Once started, the program asks for the desired maze size, up to a maximum of 35×27. Larger mazes are obviously harder to navigate (and also take longer to generate). You should probably start with something relatively small, about 5×7.

Input is through the keyboard, and comprises three commands: walk forward, face left and face right. The space bar is used to move forward, and the Apple's left and right arrow keys are used to turn. (Turning doesn't change the location of the observer, but only gives the view in another direction.) After the player has found his way out, the program gives the traditional top view of the maze and traces his perambulations. (Completely frustrated players may wish to yield to the temptation of the Escape key.)

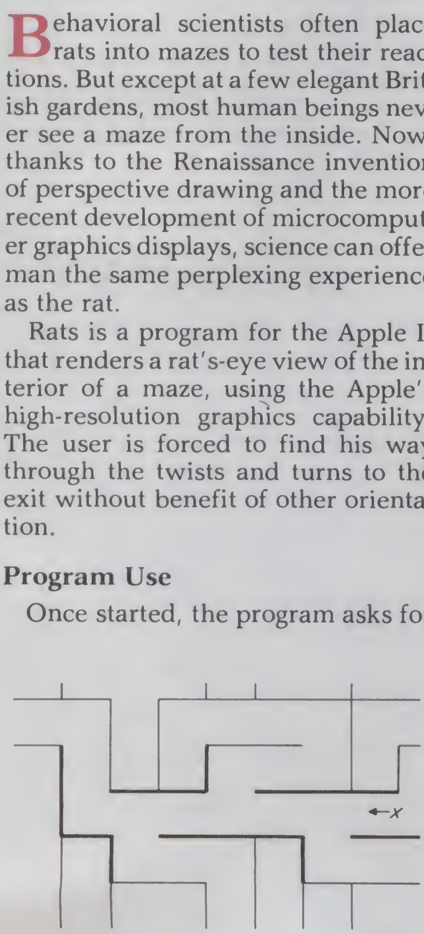


Fig. 1. Portion of a maze.

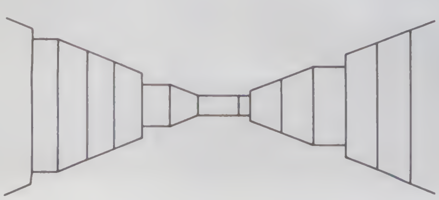


Fig. 2. Perspective view of observer in maze.

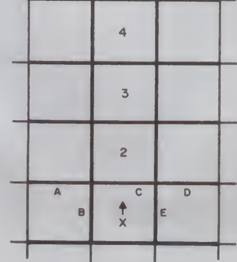


Fig. 3. Prototype maze grid.

Program Theory

Rats renders a perspective drawing of the maze from any location within it. Fig. 1 illustrates the perspective art requirement. An observer in a maze facing in the indicated direction from the spot marked X sees the view given in Fig. 2. Those walls which can be seen by the observers are emphasized in Fig. 1. Rats provides a simple algorithm for converting the description of Fig. 1 into the view in Fig. 2.

Fig. 3 gives the top view of an incomplete maze. This maze consists of a set of cells which may or may not

Address correspondence to Brian McCarson, 2922 W. Hartford Drive, Phoenix, AZ 85023.

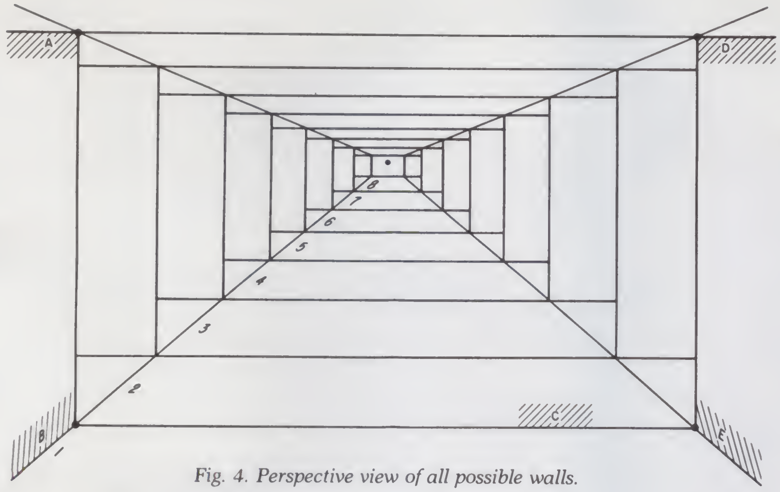


Fig. 4. Perspective view of all possible walls.

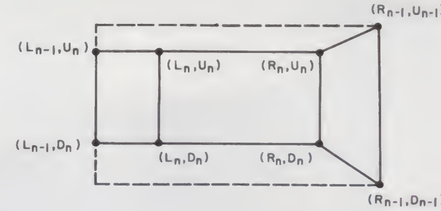


Fig. 5. Coordinates of vertices for basic walls.

have walls. An observer in the maze looks along a row of these cells. According to which walls are present, he may or may not be able to see the indicated walls. Fig. 4 shows the same set of cells and walls as seen by the observer. Notice that this view can be drawn by proceeding through the row of cells along the direction the observer is facing and drawing at each cell each of five possible walls:

- A—the back wall of the left adjacent cell.
- B—the left side wall of the cell.
- C—the back wall of the cell.
- D—the back wall of the right adjacent cell.
- E—the right side wall of the cell.

The artistic rendering of these walls is identical at each cell down the row, only smaller. (In fact, since A and D are mirror images of each other, as are B and E, there are actually only three basic shapes.)

The walls are drawn with the familiar "vanishing point" method of perspective. This means that extensions of all parallel lines moving away from the observer converge at a point (in this case, near the center of the screen). The four vertices of the center walls are the only points required to draw the image. These points are defined by four coordinate pairs composed from four values:

- U—vertical displacement of top line.
- D—vertical displacement of bottom line.
- L—horizontal displacement of left line.
- R—horizontal displacement of right line.

The side walls (B and E) are defined by two successive center walls. The back walls off to the left and right sides (A and D) are defined by the two side vertices of the center wall and the horizontal coordinate of the previous (nearer) back wall. Fig. 5 illustrates these basic coordinates. Once the nearest wall's vertices are fixed (arbitrarily according to visual aesthetics), subsequent points are calculated by simple geometric formulae by forcing convergence on the vanishing point.

The final artistic requirement is that the walls seem solid and opaque, so that hidden walls cannot be seen by "looking through" a nearer wall. The general solution to this problem in three-dimensional graphics (sometimes called the "hidden line problem") is quite complex. Fortunately, the restricted geometry of mazes immensely simplifies the task. There are, in fact, only two rules: the center wall blocks off all walls further away, and a side wall masks the back wall (if any) of the adjacent cell on that side.

The flowchart in Fig. 6 shows the basic algorithm used to draw the interior view from an arbitrary point in the maze.

Program Description

The program uses an interesting technique to increase display speed. The information in the maze variable M is highly redundant. Each element of that array indicates the walls that exist around a single cell. In theory, each cell need only record two walls (say, left and bottom) with the other two walls derived from the walls in

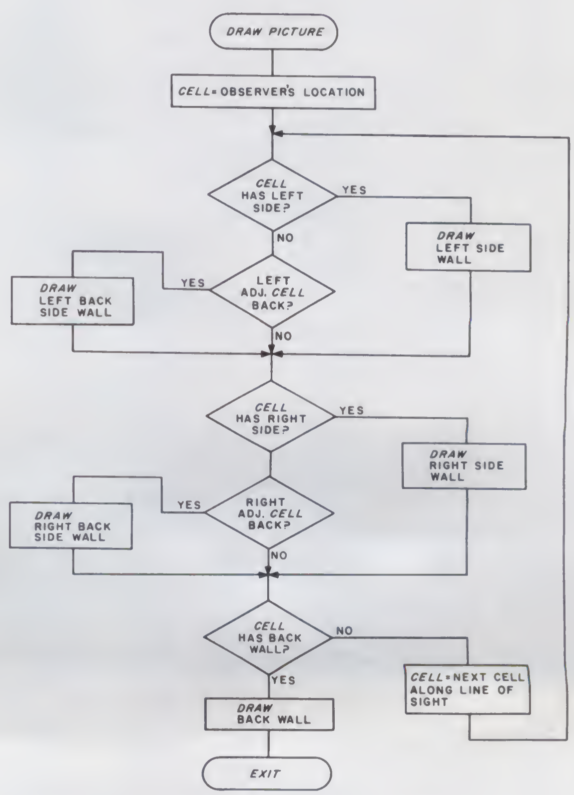


Fig. 6. Flowchart of artwork GOSUB.

North wall	M/128 MOD 2	(observer's front)
East wall	M/64 MOD 2	(observer's right)
South wall	M/32 MOD 2	(observer's rear)
West wall	M/16 MOD 2	(observer's left)

Table 1.

the adjacent cells.

In Rats, not only are all four walls recorded for each cell, but they are recorded twice. Each cell contains eight significant bits, indicating the north, east, south and west walls as:

bit:	(msb) 7	6	5	4	3	2	1	0	(lsb)
wall:	N	E	S	W	N	E	S	W	

To test for the existence of a particular wall at a cell, the program tests the wall's associated bit in the maze array M. The correct bit is isolated by a combination of division by a power of two and a MOD 2 operation, as shown in Table 1.

The division of integers by a power of two acts as a right shift (not directly available in BASIC); it removes all bits less significant than the desired bit, and positions that bit as the low order or least significant bit in the variable. The MOD 2 operation then eliminates all higher order bits, leaving the calculation result as a simple zero or one for testing in an IF statement.

This redundant encoding of the cell walls is used to assist in rapid transformation of the cell description from the absolute orientation of the maze itself to the orientation of the observer. For example, when the observer is facing east, the south wall of the cell is to the right relative to the observer. With this coding of M, the transformed cell description is achieved by a simple shift (performed in BASIC by multiplication):

observer facing	**transformation**
North	M
East	M*2
South	M*4
West	M*8

Thus, to test for a side wall on the right of a cell relative to an observer facing west, the program can test M*2/64 MOD 2. The disadvantage of this redundant coding is that the maze takes more time to generate initially.

The program is written in Apple II Integer BASIC, and uses the HIRES graphics assembly-language subroutines supplied by Apple to draw lines. The graphics are limited to a few places (GOSUBs at 200, 300, 400 and 3100, plus lines 6075 and 6275) so it can be converted to Applesoft BASIC or another machine's BASIC. The maze array variable (M) is actually a two-dimensional array; however, since Apple Integer BASIC doesn't support this, the single subscript is always calculated as M(X + Y*FX), where FX is the maximum X coordinate.

You can reduce this program's memory requirement by reducing the M array and the variable FX (lines 860 and 840). The only other functions unique to the Apple are the sound GOSUB at line 500, the CALL −936 at line 820 (which clears the character display screen) and the keyboard single character input at line 5100.■

Program listing.

```
  1 REM RATS! A 3-D MAZE PERSPECTIVE

 10 REM  SCREENPLAY BY BRUCE CARLSON

 50 POKE 75,16: REM   LOMEM
 99 GOTO 800
100 REM  DRAW 3-D VIEW AT H,V
101 REM
102 REM  H,V = CUR LOC COORDS
103 REM  S,R = FACING (F)
104 REM  A,B = LOOKING COORDS
105 N=2:A=H:B=V:FF=2 ^ (F-1)
108 CALL INIT: POKE HCLR,255
110 Z=M(A+B*FX)*FF
120 REM  LEFT SIDE
122 IF ((Z/16) MOD 2)=0 THEN 130

124 RL=-1: GOSUB 400
128 GOTO 140
130 REM  LEFT BACK
132 W=M(A+S+(B-R)*FX)*FF
134 IF ((W/128) MOD 2)=0 THEN 140

136 RL=-1: GOSUB 300
140 REM  RIGHT SIDE
142 IF ((Z/64) MOD 2)=0 THEN 150

144 RL=+1: GOSUB 400
148 GOTO 160
150 REM  RIGHT BACK
152 W=M(A-S+(B+R)*FX)*FF
154 IF ((W/128) MOD 2)=0 THEN 160

156 RL=+1: GOSUB 300
160 REM  BACK TEST
162 IF ((Z/128) MOD 2)=1 THEN 180

170 REM  NEXT LEVEL
172 N=N+1: IF N>8 THEN 190
174 A=A+R:B=B+S: IF B<2 THEN 190

178 GOTO 110
180 REM  BACK END
182 GOSUB 200
190 RETURN
200 REM  CENTER BACK SUBROUTINE
210 REM  N=DISTANCE
220 X=VX+DX(N): POKE HX,X MOD 256
    : POKE HX+1,X/256: POKE HY,
    YU(N)
225 CALL POSN
230 POKE HY,YD(N): CALL LINE
235 X=VX-DX(N): POKE HX,X MOD 256
    : POKE HX+1,X/256: CALL LINE
240 POKE HY,YU(N): CALL LINE
245 X=VX+DX(N): POKE HX,X MOD 256
    : POKE HX+1,X/256: CALL LINE
290 RETURN
300 REM  BACK SIDE SUB
310 REM  RL=BACK SIDE (-1,+1)
320 X=VX+RL*DX(N-1)
325 POKE HX,X MOD 256: POKE HX+
    1,X/256: POKE HY,YU(N)
330 CALL POSN
335 X=VX+RL*DX(N): POKE HX,X MOD
    256: POKE HX+1,X/256
340 CALL LINE
345 POKE HY,YD(N): CALL LINE
350 X=VX+RL*DX(N-1): POKE HX,X MOD
    256: POKE HX+1,X/256
360 CALL LINE
390 RETURN
400 REM  SIDE SUBR
410 REM  N=DISTANCE
415 REM  RL=WHICH SIDE (-1,+1)
420 X=VX+RL*DX(N-1): POKE HX,X MOD
    256: POKE HX+1,X/256
425 POKE HY,YU(N-1): CALL POSN
430 X=VX+RL*DX(N): POKE HX,X MOD
    256: POKE HX+1,X/256
435 POKE HY,YU(N): CALL LINE
440 POKE HY,YD(N): CALL LINE
445 X=VX+RL*DX(N-1): POKE HX,X MOD
    256: POKE HX+1,X/256
450 POKE HY,YD(N-1): CALL LINE
460 POKE HY,YU(N-1)
470 IF N>2 THEN CALL LINE
490 RETURN
500 REM :ERROR SOUND
510 SS=-16336: REM  APPLE SPEAKER
520 FOR II=1 TO 12:JJ= PEEK (SS)
    - PEEK (SS)- PEEK (SS)- PEEK
    (SS)- PEEK (SS)- PEEK (SS):
     NEXT II
530 RETURN
800 REM   HIRES GRAPHICS LOCATIONS
810 HX=800:HY=802:HCLR=812:LINE=
    3786:POSN=3761:INIT=3072
820 TEXT : CALL -936: VTAB 5: TAB
    16: PRINT "RATS!": PRINT : PRINT
    "A DIFFERENT PERSPECTIVE ON MAZE
    S."
830 POKE HCLR,7: REM  WHITE
840 VTAB 13:FX=36: INPUT "MAZE SIZE
    (H,V) ",H,V
844 IF H>2 AND H<36 AND V>2 AND
    V<28 THEN 850
846 PRINT "2<H<36 2<V<28": GOSUB
    500: GOTO 840
850 N=H*V-1:H=H+1:V=V+1
855 D=1
860 DIM M(1209),A$(10),WALK(100
    ),CUT(5),DX(8),YU(8),YD(8)
870 FOR J=1 TO V+1:JJ=J*FX:M(1+
    JJ)=4:M(H+1+JJ)=1: NEXT J
875 MX=279:MY=157:VX=MX/2:VY=MY/
    2+40:X=VX
880 REM  COMP. PERSPECTIVE PTS
890 FOR J=1 TO 8:DX(J)=X:YU(J)=
    VY-X*VY/VX:YD(J)=VY+X*(MY-VY)
    /VX:X=X*7/10: NEXT J
900 JJ=(V+1)*FX
910 FOR I=2 TO H:M(I+JJ)=8:M(I+
    FX)=2
920 FOR J=2 TO V:M(I+J*FX)=15: NEXT
    J: NEXT I
940 REM MAZE VALUES, FACING NORTH
942 REM  8=TOP WALL
943 REM  4=RIGHT WALL
```

More

Desk Main/Frame

Desk Main/Frame

LOW COST & ATTRACTIVE STYLING

- MAIN/FRAME INTEGRATED INTO FURNITURE QUALITY DESK
- ELECTRONICS PACKAGE SLIDE MOUNTED FOR EASY ACCESS
- SUPPORTS TWO 8" FLOPPY DRIVES FROM SEVERAL MANUFACTURERS (DRIVES NOT INCLUDED)
- 10 SLOT MOTHERBOARD INCLUDES CONNECTORS
- POWER SUPPLY FOR DRIVES AND CARDS
- DESK AND MAIN/FRAME AVAILABLE SEPARATELY
- MATCHING PRINTER DESK AVAILABLE

WRITE OR CALL FOR OUR BROCHURE WHICH INCLUDES OUR APPLICATION NOTE: 'BUILDING CHEAP COMPUTERS'

INTEGRAND ✓77

8474 Ave. 296 • Visalia, CA 93277 • (209) 733-9288
We accept BankAmericard/Visa and MasterCharge

AT LAST!

Mass production prices for high quality software. Buy direct and save 50%! Also available for CPM and HDOS

DATA BASE MANAGER Mod I & III $69, $149 (48K). Mod-II $199
Maintain a data base and produce reports, all without user programming. Define file parameters and report formats on-line. Key random access, fast multi-key sort, field arithmetics, audit log, label. No time-consuming overlays. 500 happy users in one year. Mod-II and 48K versions have over 50 enhancements, including 40 fields maximum. "IDM-M2 is great!" - 80-US

A/R Mod-I $69 Mod-II $149 Mod-III $69
Handles invoices, statements, aging, sales analysis, credit checking, forms input, and order entry. Unlike other accounts receivable programs, ours can be used by doctors, store managers, etc.

WORD PROCESSOR $49
Centers, justifies, indents, and numbers pages. Mod-I version features upper/lower case without hardware modification! File merge option available.

MAILING LIST Mod I & III $59, $79 (48K). Mod-II $99
The best! Compare and be selective. Includes forms input, 5-digit selection code, zip code extension, sort on any field, and multiple labels. Who else offers a report writer and merges with word processor?

INVENTORY Mod I & III $89, $109 (48K) Mod-II $149
Fast key random access. Reports include order info, performance summary, EOQ and user-specified reports. Many people have converted to our system! Next to impossible to damage the file.

GL, A/R, A/P, PAYROLL Mod-II $129 each
Integrated accounting package. 100+ page manual. As opposed to Osborne's slow binary search and 64 column screen, we use fast ISAM and 80 columns. Dual disk and TRSDOS required.

L216 $59
A cassette package of 10 business programs for Level II 16 K systems. Includes word processor and data base manager. Poker game $19

Most programs are on-line, interactive, random-access, bug-free, documented, and delivered on disks. Mod-I programs require 32K TRSDOS. We're #1 in business software—don't let our low price *fool you! Ask for our* free 20-page catalog if you're still not convinced. Compiled versions are available.

MICRO ARCHITECT, INC.
96 Dothan St., Arlington, MA 02174 ✓108

KILL SURGES LIKE LIGHTNING!

AC power line surges are destructive, can cost you money, and can't be prevented. But you can stop them from reaching your sensitive electronic equipment with a Surge Sentry.

Surge Sentry acts in picoseconds to dissipate up to a 1,000,000 W, 100μ second surge. Triggers at 10% above nominal peak voltage. Works in parallel with the power line. Is easy to install for immediate protection. No complicated wiring or special tools required.

Several different models to choose from, including an OEM version. Call or write today for a *free* brochure.

SURGE SENTRY

It'll clean up your AC

RKS ENTERPRISES, INC. ✓46
643 South 6th Street, San Jose, CA 95112
(408) 288-5565

DEALER INQUIRIES INVITED

Listing continued.

```
 944 REM  2=BOTTOM WALL
 945 REM  1=LEFT WALL
 950 REM
 951 REM  FACE  TOP RIT BOT LFT
 952 REM    N    8   4   2   1
 953 REM    E    4   2   1   8
 954 REM    S    2   1   8   4
 955 REM    W    1   8   4   2
 956 REM
 960 R=H/2+1:S=V/2+1:M(R+S*FX)=15

 970 PRINT "GENERATING MAZE.": GOSUB
     500
 980 REM  GENERATE RANDOM MAZE
 985 REM  ALGORITHM HERE FROM ROGERS
     AND STRASSBERGER
 990 FOR IWALK=1 TO N
1000 I=Z
1010 IF M(R-1+S*FX)<15 THEN 1100
1020 I=I+1:CUT(I)=1500
1100 IF M(R+(S-1)*FX)<15 THEN 1200
1110 I=I+1:CUT(I)=1600
1200 IF M(R+1+S*FX)<15 THEN 1300
1210 I=I+1:CUT(I)=1700
1300 IF M(R+(S+1)*FX)<15 THEN 1400
1310 I=I+1:CUT(I)=1800
1400 IF I=0 THEN 2000
1410 IF I#1 THEN I= RND (I)+1
1420 Q=R+S*FX
1450 GOTO CUT(I)
1500 REM   MOVE WEST
1510 M(Q)=M(Q)-(M(Q) MOD 2)
1520 R=R-1:Q=R+S*FX
1530 M(Q)=M(Q)-((M(Q)/4) MOD 2)*
     4
1550 GOTO 2500
1600 REM   MOVE NORTH
1610 M(Q)=M(Q)-((M(Q)/8) MOD 2)*
     8
1620 S=S-1:Q=R+S*FX
1630 M(Q)=M(Q)-((M(Q)/2) MOD 2)*
     2
1650 GOTO 2500
1700 REM   MOVE EAST
1710 M(Q)=M(Q)-((M(Q)/4) MOD 2)*
     4
1720 R=R+1:Q=R+S*FX
1730 M(Q)=M(Q)-(M(Q) MOD 2)
1750 GOTO 2500
1800 REM   MOVE SOUTH
1810 M(Q)=M(Q)-((M(Q)/2) MOD 2)*
     2
1820 S=S+1:Q=R+S*FX
1830 M(Q)=M(Q)-((M(Q)/8) MOD 2)*
     8
1850 GOTO 2500
1900 REM  SCAN FOR UNUSED PT.
2000 IF D=-1 THEN 2100
2010 IF R#H THEN 2300
2020 IF S#V THEN 2200
2030 R=2:S=2: GOTO 2400
2100 IF R#2 THEN 2300
2110 IF S#V THEN 2200
2120 R=H:S=2: GOTO 2400
2200 S=S+1:D=-D: GOTO 2400
2300 R=R+D
2400 IF M(R+S*FX)=15 THEN 2000
2450 GOTO 1000
2500 NEXT IWALK
3000 MH=H:MV=V
3001 REM  RANDOM PT AT SOUTH AS START
     ING  POINT
3005 I= RND (MH-1)+2
3010 M(I+1*FX)=0
3015 II=I+2*FX:M(II)=M(II)-((M(II)
     /8) MOD 2)*8
3020 H= RND (MH-1)+2
3030 H1=H:V1=V
3050 GOSUB 500
3080 VTAB 23: TAB 12: PRINT "MAZE COM
     PLETED.": GOSUB 500
3090 GOTO 3600
3100 REM  DISPLAY TOP VIEW OF MAZE
3140 REM  CALC DISPLAY SCALE
3150 HZ=278/MH:VZ=158/MV
3200 CALL INIT: POKE HCLR,255
3231 POKE HX,1+HZ: POKE HX+1,0: POKE
     HY,1+VZ: CALL POSN
3232 POKE HY,MV*VZ+1: CALL LINE
3240 FOR J=1 TO MV
3250 FOR I=2 TO MH
3260 N=M(I+J*FX):X=I*HZ+1:Y=J*VZ+
     1
3290 IF ((N/2) MOD 2)=0 THEN 3400

3299 REM DRAW SOUTH WALL
3300 POKE HX,X MOD 256: POKE HX+
     1,X/256: POKE HY,Y: CALL POSN
3330 POKE HX,(X-HZ) MOD 256: POKE
     HX+1,(X-HZ)/256: CALL LINE
3400 IF ((N/4) MOD 2)=0 THEN 3500

3405 REM  DRAW EAST WALL
3410 POKE HX,X MOD 256: POKE HX+
     1,X/256: POKE HY,Y: CALL POSN
3450 POKE HY,(Y-VZ): CALL LINE
3500 NEXT I
3510 NEXT J
3530 REM  MARK RAT'S POSITION
3535 X=H*HZ-1:Y=V*VZ-1
3540 POKE HX,(X+1) MOD 256: POKE
     HX+1,(X+1)/256: POKE HY,Y+1
     : CALL POSN
3545 POKE HX,(X-HZ+2) MOD 256: POKE
     HX+1,(X-HZ+2)/256: POKE HY,
     Y-VZ+2: CALL LINE
3550 POKE HY,Y: CALL POSN
3555 POKE HX,(X+1) MOD 256: POKE
     HX+1,(X+1)/256: POKE HY,Y-VZ+
     2: CALL LINE
3590 RETURN
3600 REM  ADD REDUNDANT BITS
3610 FOR X=1 TO MH: FOR Y=1 TO MV
3620 I=X+Y*FX:M(I)=M(I)+M(I)*16
3650 NEXT Y: NEXT X
5000 REM  PLAY
5001 REM  RANDOM INITIAL DIRECTION
5005 F= RND (4)+1: GOTO 5010+F
5010 F=1:R=0:S=-1
5011 R=0:S=-1: GOTO 5020
5012 R=+1:S=0: GOTO 5020
5013 R=0:S=+1: GOTO 5020
5014 R=-1:S=0
5020 PRINT "LEFT ARROW <- TO TURN LEF
     T."
5022 PRINT "RIGHT ARROW -> TO TURN RI
     GHT."
5024 PRINT "SPACE BAR TO GO FORWARD."

5040 GOSUB 100
5090 REM  GET KEYSTROKE
5100 K= PEEK (-16384): IF K<128 THEN
     5100
5110 POKE -16368,0
5120 IF K=136 THEN 5200: REM  <-
5130 IF K=149 THEN 5300: REM  ->
5140 IF K=160 THEN 5400: REM  SPACE
5150 IF K=155 THEN 5700: REM  ESC
5160 GOSUB 500: GOTO 5020
5200 REM  LEFT <-
5210 F=F-1: IF F<1 THEN F=4
5220 GOTO 5350
5300 REM  RIGTH ->
5310 F=F+1: IF F>4 THEN F=1
5350 GOTO 5360+F
5361 R=0:S=-1: GOTO 5370
5362 R=1:S=0: GOTO 5370
5363 R=0:S=+1: GOTO 5370
5364 R=-1:S=0: GOTO 5370
5370 GOTO 5500
5400 REM  FORWARD
5410 Z=M(H+V*FX)
5415 T=Z*2 ^ (F-1):T=(T/128) MOD
     2
5430 IF T=0 THEN 5450
5435 REM  BUMPED INTO WALL
5440 GOSUB 500: GOTO 5100
5450 NM=NM+1: VTAB 22: TAB 34: PRINT
     "MOVE ";NM
5455 IF NM<100 THEN WALK(NM)=F
5460 H=H+R:V=V+S
5480 IF V<2 THEN GOTO 6000
5500 GOSUB 100
5590 GOTO 5100
5700 REM  ESCAPE
5710 GOSUB 3100
5720 GOTO 5100
6000 PRINT : PRINT "CONGRATULATIONS,
     YOU'RE OUT IN ";NM;" STEPS."

6010 GOSUB 500
6020 V=V1:H=H1: GOSUB 3100
6030 REM  DRAW PATH WALKED
6050 X=H*HZ-HZ/2:Y=V*VZ-VZ/2
6075 POKE HX,X MOD 256: POKE HX+
     1,X/256: POKE HY,Y: CALL POSN
6100 FOR N=1 TO NM
6110 IF N>100 THEN 6300
6125 F=WALK(N)
6150 IF F=1 THEN V=V-1
6175 IF F=2 THEN H=H+1
6200 IF F=3 THEN V=V+1
6225 IF F=4 THEN H=H-1
6250 X=H*HZ-HZ/2:Y=V*VZ-VZ/2
6275 POKE HX,X MOD 256: POKE HX+
     1,X/256: POKE HY,Y: CALL LINE
6300 NEXT N
6999 END
```


DIMENSION 3
COLOR GRAPHICS
Takes your PET™, CBM™ or TRS-80™ into the future
Our VDP color board features 256 x 192 pixel resolution, 15 exciting colors on 5 planes producing 3-dimensional graphics and true animation, standard composite video output to either T.V. set or monitor (through optional R.F. Modulator)
Plus 3 channel digitally synthesized sound and 16k onboard RAM, driver software is available to maximize user flexibility. Install Dimension 3 in your present computer within minutes yourself. No technical expertise required. Dimension 3 from Dimensional Design Inc. is available through Computer City Canada (dealer enquiries welcomed).
PET, CBM are registered TM's of Commodore International. TRS-80 is a registered TM of Tandy Corporation. Dimension 3 is a registered TM of DDI Dimensional Design Inc. Artist's representation.
Introductory price offer (PET-CBM) $595.00 U.S. TRS-80 $649.00 U.S.
MAIL TO: Computer City, Canada Inc. 1353 Portage Avenue, Winnipeg, Canada R3G 0V5 Phone: (204) 786-3383
✓224
Name .... Address ....
Prov./State .... Postal/Zip Code .... Country ....
Please send me ( ) color cards. My system is TRS-80 ( ) at $649.00 U.S.
original PET ( ) or CBM ( ) at $595.00 U.S.
Enclosed please find $ ....
Master Charge # ....
Visa # ....
Signature ....
Make cheques or money orders payable to Computer City, Canada Inc. Master Charge or Visa add 5%.

Your vehicle for com
The Challenger 8P DF.

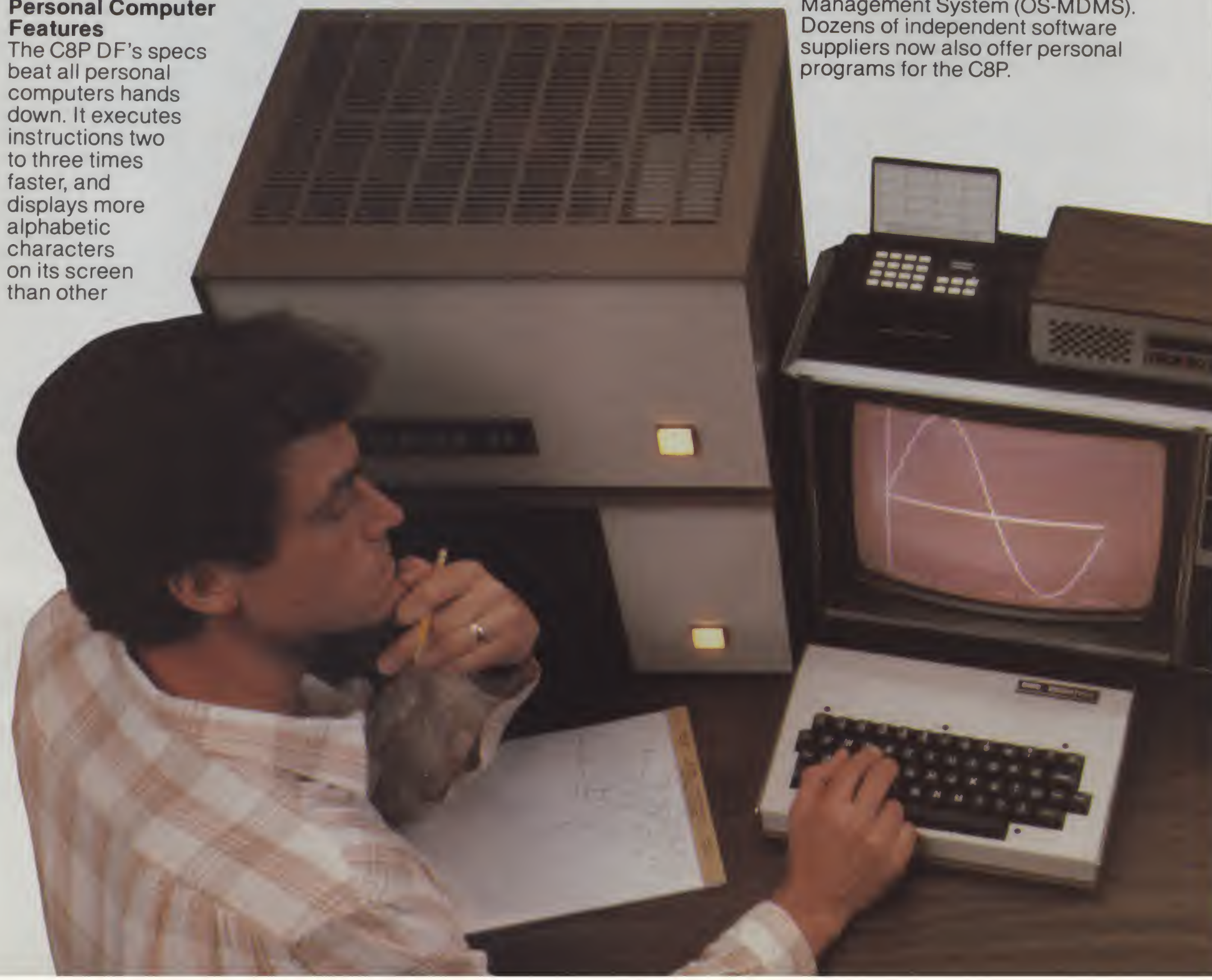
The general purpose microcomputer was first introduced as a computer for hobbyists and experimenters. However, as the industry has grown, microcomputers have become specialized for personal use or for small business use. There is virtually no computer for the serious experimenter with one important exception, the Ohio Scientific Challenger 8P.

The C8P is unique in that it incorporates the features of state-of-the-art personal computers, with the memory and disk storage capacity of business computers, along with the "mainframe" bus architecture and open ended expansion capability of industrial control computers.

Personal Computer Features
The C8P DF's specs beat all personal computers hands down. It executes instructions two to three times faster, and displays more alphabetic characters on its screen than other models. It has upper and lower case and graphics in 16 colors. The C8P's *standard* I/O capabilities are far more extensive than any other computer, with joystick and keypad interfaces, sound output, an 8-bit D/A converter, 16 parallel I/O lines, modem and printer interfaces, AC remote control and security monitor interfaces and a universal accessory port that accepts a prom blaster, 12-bit analog I/O module, solderless prototyping board and more.

Ohio Scientific offers a large library of personal applications programs, including exciting action games such as Invaders and Star Trek, sports simulations, games of logic and educational games, personal applications such as biorhythms, calorie counter, home programs such as checking and savings account balancers and a home budgeter just to name a few. A new Plot BASIC makes elaborate animations easy, and music composition program allows you to play complex multi-part music through the computers DAC.

At the systems level the machine comes standard with OS-65D, an advanced disk operating system with Microsoft BASIC and an interactive Assembler Editor. Optional software includes UCSD PASCAL and FORTRAN and an Information Management System (OS-MDMS). Dozens of independent software suppliers now also offer personal programs for the C8P.



puter explorations.

Business Computer Features
The C8P DF utilizes dual 8″ floppy disk drives which store up to eight times as much information as personal computer mini-floppies, and an available double-sided option expands capacity to 1.2 megabytes of on-line storage. The C8P DF is compatible with Ohio Scientific's business computer software, including OS-65U an advanced operating system, and an Information Management System (OS-DMS) with supplementary inventory, accounting, A/R-A/P, payroll, purchasing, estimation, educational grading and financial modeling packages. The system also supports word processing (WP-3) and a fully integrated small business accounting system (OS-AMCAP V1.6). The C8P DF's standard modem and printer ports accept high-speed matrix printers and word-processing printers directly.

Home Control and Industrial Control
The C8P DF has the most advanced home monitoring and control capabilities ever offered in a computer system. It incorporates a real time clock and a unique FOREGROUND/BACKGROUND operating system which allows the computer to function with normal BASIC programs, at the same time it is monitoring external devices. The C8P DF comes standard with an AC remote control interface, which allows it to control a wide range of AC appliances and lights remotely, without wiring, and an interface for home security systems which monitors fire, intrusion, car theft, water levels and freezer temperature, all without messy wiring. In addition, the C8P DF can accept Ohio Scientific's Votrax voice I/O board and/or Ohio Scientific's new universal telephone interface (UTI). The telephone interface connects the computer to any telephone line. The computer system is able to answer calls, initiate calls and communicate via touch-tone signals, voice output or 300 baud modem signals. It can accept and decode touch-tone signals, 300 baud modem signals and record incoming voice messages. These features collectively give the C8P DF capabilities to monitor and control home functions with almost human-like capabilities.

For process control applications, a battery back up calendar clock with automatic computer restart capabilities is available. Ohio Scientific's unique accessory ports allow the connection of a nearly unlimited number of 48 line parallel I/O cards and 12-bit high speed instrumentation quality analog I/O modules to the computer by inexpensive 16-pin ribbon cables.

Exploring New Frontiers
Ohio Scientific's vocalizer software processes normal BASIC print statements with conventional spellings and speaks them clearly in real-time on computers equipped with the UTI (CA-15B or CA-14A). This voice output capability, combined with the C8P's remote control, remote sensing, telephone interface capabilities and reasonable cost open up new frontiers for computer applications.

Documentation
The C8P DF is not a beginner's computer and doesn't come with beginner's documentation. However, Ohio Scientific does offer detailed documentation on the computer which is meaningful for experts, including a Howard Sams produced hardware service manual that includes detailed block diagrams, schematics, parts placement diagrams and parts lists. Ohio Scientific is now also offering fully documented Source Code in machine readable form for OS-65D, the Challenger 8P's operating system allowing experimenters and industrial users to customize the system to their specific applications.

What's Next?
Ohio Scientific is working on a speech recognizer to complement the UTI system, with a several hundred word vocabulary. The company is also developing an 8 megabyte low-cost, add-on hard disk for use in conjunction with natural language parsing to further advance the state-of-the-art in small computers. The modular bus architecture of the C8P assures system owners of being able to make use of these new developments as they become available just as the owner of a 1976 vintage Challenger can directly plug in voice output, the UTI and other current state-of-the-art OSI products.

The C8P DF with dual 8″ floppies, BASIC and two operating systems costs about $3000, only slightly more than you would pay for a dual mini-floppy equipped personal computer with only a fraction of the capabilities of the C8P.

For more information and the name of the dealer nearest you, call 1-800-321-6850 toll free.

OHIO SCIENTIFIC
a M/A-COM Company
1333 SOUTH CHILLICOTHE ROAD
AURORA, OH 44202 • (216) 831-5600

361

You may be shocked or entertained by this random, fun-sentence generator for the Apple II Plus. Definitely not for the prudish.

Micros Say The Darn'dest Things

By Michael Grady

In recent years, the study of grammar has been revolutionized by the application of such principles as trees, nodes and branching. These are terms familiar to the computer programmer. In fact, the impetus for one kind of linguistic analysis—developed chiefly by MIT's Noam Chomsky and implemented by many others—was computer analysis and computer text-generation of English sentences.

In many ways, the original concept of Chomsky's generative-transformational grammar has not proven out. But the recreational programmer can still have some fun with it, if he acknowledges the theory's limitations when applied to a microcomputer.

This random fun-sentence program is essentially a variety of random "branching tree" programming. A very limited vocabulary of English words is subclassified into elementary linguistic categories. The computer generates a basic sentence (S$) by including mandatory, higher-level, categories in S$, and randomly choosing to include or not various other, optional, categories.

In actual practice, a high-level category may be required, as for AV$ in line 1060, but subsequent randomization may reduce the high-level category to the null string, as in lines 1070–1100. The actual vocabulary items that are "plugged into" S$ are randomly chosen as well, subject to certain constraints discussed later on.

The program is written in Applesoft BASIC and should run without much modification on computers with similar dialects. If you want to include a larger vocabulary, you'll have to dimension each linguistic category array. For arrays of up to 11 elements, however, Applesoft doesn't require dimensioning, and to ease the programming burden, I kept the items for each fundamental category to ten or fewer. The program occupies about 4K of memory.

The program doesn't cover all linguistic rules in English, so some of the "sentences" it generates are only marginally grammatical. Others are "grammatical," in some sense of the term, but include odd word combinations. But that's where the fun lies.

Six sentences appear at a time, and the program loops endlessly as long

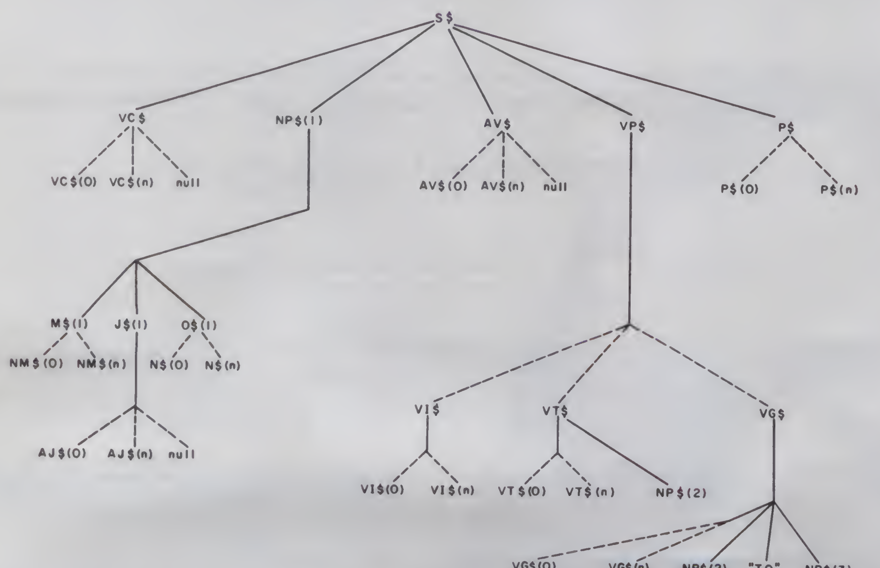


Fig. 1. Random sentence tree. Solid lines indicate mandatory inclusion under a node. Broken lines indicate randomized choices on a given level. The structure for NP$(2) and NP$(3) is identical to that given for NP$(1), but, of course, with the appropriate numerical subscripts.

Michael Grady is the head of the information consulting firm Michael Grady Associates, 332 South Juniper St., Suite 204A, Escondido, CA 92025. Before launching a career as a professional writer, he was a university professor of linguistics.

as you press "R" each time you're given a prompt. Rarely will you see the same sentence twice. You can extend the vocabulary and categories as you wish, to get even longer and more "creative" sentences, with even less likelihood of duplication. And depending on which words you later decide to drop into the program on your own, you can have either a highly moral sentence generator, or one that would make a stevedore blush.

I'm deliberately not including any sample runs with the listing, so as not to take away the element of shock, surprise and (I hope!) hilarity as the sentences appear on your screen.

A Few Structural Details

The grammatical categories set up in the program are as follows:

- VC$(n)—The $ symbol in Applesoft (and other dialects and languages) indicates an alphanumeric string. The number in parentheses indicates a specific element in an array. Modify according to your computer's dialect, if you don't run Applesoft.

VC is a mnemonic for "vocative," or what some linguists term a "call." It's an attention-getting phrase (or sometimes even a full sentence) that a speaker utters before entering into the heart of his topic. The program requires a VC$, but in a subsequent operation may assign "nullness" to VC$, on a random basis.

- NM$—A mnemonic for the term "noun marker," representing a class of English function-words which don't carry a lot of dictionary meaning but serve as linguistic markers that a noun is coming up. They're also called determiners, articles and possessive pronouns. They all tend to function alike grammatically. An intermediate category of noun marker is termed M$(n) in the program.
- N$—Noun, of course. An intermediate string in the program is termed O$(n) (from nOun. Ns were getting a lot of use, and I had to use an interior letter in this word as a mnemonic).
- NP$—Noun phrase, a high-level category. This consists of a mandatory noun marker string, an optional adjective string and a mandatory noun (O$[n]) string.
- VP$—Verb phrase, another high-level category. A VP$ consists of a verb of some sort—intransitive, transitive or give—plus whatever might follow the transitive or give verb (at least one NP$).
- VI$—Verb intransitive. Those English verbs that don't require a fol-

Listing 1. A random, fun-sentence generator program in Applesoft BASIC.

```
10  HOME
20  REM  RANDOM SENTENCE GENERATOR
30 VC$(0) = "LO! "
40 VC$(1) = "MY! "
50 VC$(2) = "HARK! "
60 VC$(3) = "I CAN'T BELIEVE IT! "
70 VC$(4) = "OH, MY LORD! "
80 VC$(5) = "GOODNESS, GRACIOUS, ME: "
90 VC$(6) = "IT'S A FACT! "
100 VC$(7) = "ALAS, "
110 VC$(8) = "OH, "
120 VC$(9) = "ASTOUNDING! "
130 NM$(0) = "THIS "
140 NM$(1) = "THAT "
150 NM$(2) = "HER "
160 NM$(3) = "EACH "
170 NM$(4) = "MY "
180 NM$(5) = "THE "
190 NM$(6) = "OUR "
200 NM$(7) = "HIS "
210 NM$(8) = "YOUR "
220 NM$(9) = "THE "
230 N$(0) = "BOY "
240 N$(1) = "GIRL "
250 N$(2) = "PET SNAKE "
260 N$(3) = "PLUMBER "
270 N$(4) = "POODLE "
280 N$(5) = "DRAGON "
290 N$(6) = "FROG "
300 N$(7) = "BEETLE "
310 N$(8) = "NAUGHTY PART "
320  REM  FROM MONTY PYTHON'S TV SERIES!
330 N$(9) = "CHIPMUNK "
340 VI$(0) = "GROWLED "
350 VI$(1) = "DRANK TO A STUPOR "
360 VI$(2) = "EVAPORATED "
370 VI$(3) = "STAGGERED "
380 VI$(4) = "BOUNCED "
390 VI$(5) = "TWIRLS "
400 VI$(6) = "BUZZES "
410 VI$(7) = "LAUGHED "
420 VI$(8) = "EXPLODED "
430 VI$(9) = "COUGHED "
440 VB$(0) = "IS "
450 VB$(1) = "WAS "
460 VB$(2) = "APPEARS "
470 VB$(3) = "SEEMED "
480 VB$(4) = "SOUNDS "
490 VB$(5) = "LOOKED "
500 VB$(6) = "APPEARED "
510 VB$(7) = "REMAINED "
520 VB$(8) = "STAYS "
530 VB$(9) = "SEEMS "
540 VT$(0) = "PUZZLED "
550 VT$(1) = "SMELLS "
560 VT$(2) = "GOOSED "
570 VT$(3) = "CURSED "
580 VT$(4) = "DEMANDS "
590 VT$(5) = "SHOT "
600 VT$(6) = "WASHED "
610 VT$(7) = "KISSED "
620 VT$(8) = "MOUNTED "
630 VT$(9) = "BROILED "
640 VG$(0) = "GIVES "
650 VG$(1) = "GIVES "
660 VG$(2) = "SENT "
670 VG$(3) = "BROUGHT "
680 VG$(4) = "OFFERED "
690 VG$(5) = "HANDS "
700 VG$(6) = "NAILED "
710 VG$(7) = "THREW "
720 VG$(8) = "DONATED "
730 VG$(9) = "SENDS "
740 AV$(0) = "DELIBERATELY "
750 AV$(1) = "DESPERATELY "
760 AV$(2) = "BENIGNLY "
770 AV$(3) = "OPENLY "
780 AV$(4) = "WITH CONSUMMATE GRACE AND SKILL "
790 AV$(5) = "CHARMINGLY "
800 AV$(6) = "SAVAGELY "
810 AV$(7) = "WISTFULLY "
820 AV$(8) = "DELICATELY "
830 AV$(9) = "SERENELY "
840 AJ$(0) = "BRAVE "
850 AJ$(1) = "GORGEOUS "
860 AJ$(2) = "FURRY "
870 AJ$(3) = "AUSTERE "
```

More

Listing 1 continued.

```
880 AJ$(4) = "DAINTY "
890 AJ$(5) = "AWE-INSPIRING "
900 AJ$(6) = "SURLY "
910 AJ$(7) = "MAGNIFICENT "
920 AJ$(8) = "IMMACULATE "
930 AJ$(9) = "SCIENTIFIC "
940 P$(0) = "."
950 P$(1) = "!"
960 P$(2) = "?"
970 P$(3) = "!"
980 P$(4) = "."
990 P$(5) = "."
1000 P$(6) = "."
1010 P$(7) = "."
1020 P$(8) = "!"
1030 P$(9) = "?"
1040  DEF  FN R(X) =  INT ( RND (2) * (10))
1050  FOR N = 0 TO 5
1060 S$ = VC$ + NP$(1) + AV$ + VP$ + P$
1070 X =  FN R(X)
1080  IF X =  < 4 THEN VC$ = "": GOTO 1110
1090 X =  FN R(X)
1100 VC$ = VC$(X)
1110 X =  FN R(X)
1120  IF X =  < 4 THEN J$(1) = "": GOTO 1150
1130 X =  FN R(X)
1140 J$(1) = AJ$(X)
1150 X =  FN R(X)
1160 M$(1) = NM$(X)
1170 X =  FN R(X)
1180  IF X = 8 THEN X = 2
1190 O$(1) = N$(X)
1200 NP$(1) = M$(1) + J$(1) + O$(1)
1210 X =  FN R(X)
1220  IF X =  < 4 THEN AV$ = "": GOTO 1250
1230 X =  FN R(X)
1240 AV$ = AV$(X)
1250 X =  FN R(X)
1260  IF X =  < 4 THEN X =  FN R(X):VP$ = VI$(X): GOTO 1570
1270 X =  FN R(X)
1280  IF X =  < 4 THEN X =  FN R(X):VT$ = VT$(X):T = 1: GOTO 1300
1290 X =  FN R(X):VG$ = VG$(X)
1300 X =  FN R(X)
1310  IF X =  < 4 THEN J$(2) = "": GOTO 1350
1320 X =  FN R(X)
1330 J$(2) = AJ$(X)
1340  IF J$(2) = J$(1) THEN J$(2) = "": GOTO 1320
1350 X =  FN R(X)
1360 M$(2) = NM$(X)
1370  IF M$(2) = M$(1) THEN M$(2) = "": GOTO 1350
1380 X =  FN R(X)
1390 O$(2) = N$(X)
1400  IF O$(2) = O$(1) THEN O$(2) = "": GOTO 1380
1410 NP$(2) = M$(2) + J$(2) + O$(2)
1420  IF T = 1 THEN T = 0:VP$ = VT$ + NP$(2): GOTO 1570
1430 X =  FN R(X)
1440  IF X =  < 4 THEN J$(3) = "": GOTO 1480
1450 X =  FN R(X)
1460 J$(3) = AJ$(X)
1470  IF J$(3) = J$(1) OR J$(3) = J$(2) THEN J$(3) = "": GOTO 145
     0
1480 X =  FN R(X)
1490 M$(3) = NM$(X)
1500  IF M$(3) = M$(1) OR M$(3) = M$(2) THEN M$(3) = "": GOTO 148
     0
1510 X =  FN R(X)
1520  IF X = 8 THEN X = 7
1530 O$(3) = N$(X)
1540  IF O$(3) = O$(1) OR O$(3) = O$(2) THEN O$(3) = "": GOTO 151
     0
1550 NP$(3) = M$(3) + J$(3) + O$(3)
1560 VP$ = VG$ + NP$(2) + "TO " + NP$(3)
1570 X =  FN R(X)
1580 P$ =  CHR$ (8) + P$(X)
1590 L =  LEN (S$): IF L =  < 40 THEN A$ = S$: GOTO 1740
1600  FOR B = 39 TO 1 STEP  - 1
1610 T$ =  MID$ (S$,B,1)
1620  IF T$ =  CHR$ (32) THEN B$ =  RIGHT$ (S$,L - B):A$ =  LEFT$
     (S$,B): GOTO 1640
1630  NEXT B
1640 L =  LEN (B$): IF L =  < 40 GOTO 1740
1650  FOR C = 38 TO 1 STEP  - 1
1660 T$ =  MID$ (B$,C,1)
1670  IF T$ =  CHR$ (32) THEN C$ =  RIGHT$ (B$,L - C):B$ =  LEFT$
     (B$,C): GOTO 1690
1680  NEXT C
1690 L =  LEN (C$): IF L =  < 40 GOTO 1740
1700  FOR D = 38 TO 1 STEP  - 1
1710 T$ =  MID$ (C$,D,1)
```

More →

lowing noun or adjective.

●VB$—Verb "become." "Become" is a commonly-occurring verb of this subclass. A variety of the linking verb. In this program, instances of "to be" are also included in the category VB, though on other linguistic grounds "to be" belongs in its own category.

●VT$—Verb transitive. A verb whose expressed action carries over to a single noun object.

●VG$—Verb give. A subclass of verb which takes indirect objects as well as direct objects. Named after "give," probably the most frequently-occurring verb in this category.

●AV$—Adverb. All in the program are manner adverbs, because they lead to funnier results and are simpler to handle linguistically. (Manner adverbs answer the question "How?" Remember?)

●AJ$—Adjective. Noun modifiers. An intermediate string in the program is called J$(n).

●PS$—Punctuation string. Periods, question marks, exclamation marks, all selected randomly and not always appropriately.

Some Technical Remarks

Line 1040 [1040 DEF FN R(X) = INT (RND(2)*(10))] takes advantage of the Applesoft provision for creating a user-defined function. In this case, I've defined function R of X as randomly generating an integer between 0 and 9. The number in parentheses after RND must be positive and larger than zero, for X to be randomly generated each time RND is called. The number actually returned is a decimal, and multiplying it by 10 moves the decimal one place to the right, of course. Then the INT function chops off the decimal altogether.

For an expanded vocabulary, the programmer can increase the power of 10 (10 to the second yields values for X ranging from 0 through 99). The only reason I have "10" in parentheses is to preserve it in pristine purity throughout the line renumbering program I've used. Without the parentheses, "10" has come out as various interesting but useless numerals. Line 1070 gives the syntax for the use of FN R, as previously defined.

Line 1180 excludes one noun from occurring in subject position, where it's not compatible with most verbs in the program. If the value returned for X by FN R is 8, then X is revalued as 2. The same thing happens in line 1520, to exclude the same noun from

NORTH STAR HORIZON:	
HRZ-2-32K-D-Factory ASM	$2275
HRZ-2-32K-Q-Factory ASM	$2675
64K DD or Q Also Available	
HDS-18-F	$4449
Medical-Dental SYS	$2500
HP-85A SPECIAL	$2795
HP-83	2590
We carry the complete HP line	
COMMODORE (PET):	
2001-32K-BorN Keyboard	$1090
2001-16K	$ 859
8032 (80 Column Screen)	$1599
2040 Dual Floppy Drive	$1090
8050 Dual Floppy (1MEG)	$1599
2022 Tractor Printer	$ 749
Word PRO IV	$ 290
VISICALC	$ 195
APPLE II PLUS Call For Price!	
ATARI 800	$ 849
TI 99/4 Console & Monitor	$ 990
INTERTEC SUPERBRAIN:	
32K RAM	$2595
64K RAM	$2695
PRINTERS	
Letter Quality	
NEC 5510 or 5530	$2550
NEC 5520 KSR	$2950
DIABLO 630	$2390
C ITOH	$1690
Dot Matrix	
CENTRONICS 730	$ 599
CENTRONICS 737-1	$ 799
CENTRONICS 799	$ 999
EPSON MX-80	$ 599
PAPER TIGER 460G	$1250
PAPER TIGER 445G	$ 820
DISPLAY TERMINALS	
HAZELTINE 1420	$ 949
HAZELTINE 1500	$ 999
INTERTUBE III	$ 775
TELEVIDEO 920C	$ 849

Complete sales and service since 1977. Most items in stock, prices are subject to change. Visa and Master Charge Welcome.

Multi Business Computer Systems Inc.

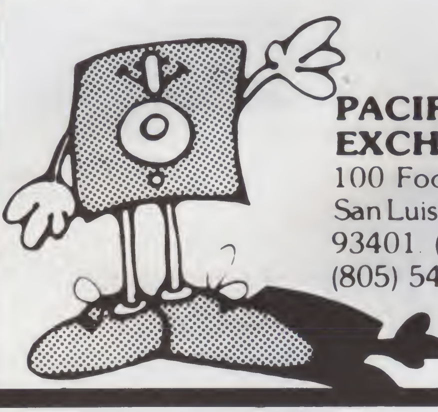
Mon. thru Fri. 9:00 - 6:00
Sat. 9:30 - 3:00

28 Marlborough St. Portland, CT 06480
(203) 342-2747 TWX/TELEX 710-428-6345

✓81

Dysan CORPORATION

Solve your disk problems, buy 100% surface tested Dysan diskettes. All orders shipped from stock, within 24 hours. Call toll FREE (800) 235-4137 for prices and information. Visa and Master Card accepted. All orders sent postage paid.



PACIFIC EXCHANGES
100 Foothill Blvd.
San Luis Obispo, CA
93401. (In Cal. call
(805) 543-1037.)

✓274

Save With

Service Technologies

"Simply the Best"

ADD LOWER CASE TO ANY CENTRONICS PRINTER! . . . $89.00

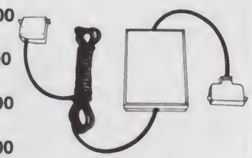
ADD MOTOR CONTROL TO YOUR 779! $95.00

Our Conversion Kit I gives all Centronics printers the ability to print upper and lower case.....$89.00.
Optional customer defined character sets available
Our Conversion Kit II turns your motor on and off automatically. Increases the life of your printer:.....$95.00.
Both kits require NO soldering, etchcuts, software change, interface or program modifications. Installs in minutes with just a screw driver

We are the original Centronics add-on company. Don't be fooled by immatations!

NOW USE YOUR CENTRONICS PARALLEL PRINTER ON ANY COMPUTER WITH A SERIAL PORT!
AVAILABLE IN FOUR MODELS, ALL WITH SWITCHABLE BAUD RATES

CSP - 100
NO BUFFER $175.00
CSP - 200
256 CHARACTER BUFFER $225.00
CSP - 200X
X ON/X OFF PROTOCOL $340.00
CSP - 2000
2048 CHARACTER BUFFER $395.00



And more......

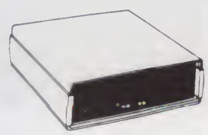
C-488 IEEE PARALLEL "LISTEN ONLY" TO CENTRONICS INTERFACE

- Input IEEE 488 GPIB, Hewlett Packard HPIB, or PET compatible
- Switchable device address
- Disable switch for address and parallel commands $125.00

PSI PARALLEL SWITCH ALLOWING TWO DEVICES TO SHARE ONE LINE PRINTER

- Two parallel input ports - one parallel output port
- Manual or remote switching control $125.00

SDP-1
A USER PROGRAMABLE SERIAL DATA PROCESSOR WITH THE CAPABILITY OF COMBINING MULTIPLE SERIAL DATA OPERATIONS



- Z-80 Microprocessor - controlled
- Buffer allocation, synchronous or asynchronous operation including transparent or non transparent mode.
- Code, word structure and buad rate conversions Ex. ASCII to EBCID or ASCII to BAUDOT
- Handles baud rates of 50-19,200 b.p.s
- Buffer size expandable from 4K - 64K RAM memory
- Audio cassette interface
- Independent and separately controlled input and output ports.

OUR QUALITY, SERVICE, WARRENTY AND PRICE CAN'T BE BEAT

To order please send check or money order to;

Service Technologies
32 Nightingale Rd.
Nashua, N.H. 03062
(603) 883-5369

✓208

master charge VISA

Visa and Master Card accepted (include signature expiration date and phone #)
phone orders and inquirys always welcome

Listing 1 continued.

```
1720  IF T$ =  CHR$ (32) THEN D$ =  RIGHT$ (C$,L - D):C$ =  LEFT$
     (C$,D): GOTO 1740
1730  NEXT D
1740  PRINT A$
1750  IF B$ <  > "" THEN  PRINT B$
1760  IF C$ <  > "" THEN  PRINT C$
1770  IF D$ <  > "" THEN  PRINT D$
1780  PRINT :B$ = "":C$ = "":D$ = ""
1790  NEXT N
1800  VTAB 23: PRINT " <<'E' TO END; 'R' TO RE-RUN.>>": GET X$
1810  IF X$ = "E" THEN  END
1820  IF X$ = "R" THEN  HOME : GOTO 1050
1830  GOTO 1800
```

indirect object position. But here X is assigned the value 7.

There's a method in this, by the way. I suspected the random numbers returned by FN R were really not all that random, and tested my Apple on runs of 1000 and 10,000 separate randomly generated numbers. In both runs there was a significantly lower return of "2" and "7" than one would expect on a purely randomly generated basis. Lines 1180 and 1520 partially compensate for the glitch.

Line 1080 is the first use of randomization in the program to force an either/or choice. If the random number generated lies within the lower half of the range 0–9, one sequence occurs—in this case, the nulling of VC$. If the test fails, a new random number is called for, and a different sequence occurs—here the assignment of a specific array element to VC$. Other array items are assigned to higher-level categories in a similar manner.

Line 1340 is also repeated in spirit throughout the rest of the program. It is a conditional loop that keeps the same word from being used more than once in a given sentence.

Line 1580 includes CHR$(8) as the first element of P$. CHR$(8) is backspace, and places the punctuation mark that follows just after the last word of the sentence. All vocabulary items in lines 30 through 930 include a trailing space as a word separator, and CHR$(8) compensates for what would otherwise be a noticeable gap at the end of a sentence.

Lines 1590–1730 format the screen display for Apple's 40-character line. The program lines assure that only full words will appear on each screen line (i.e., wrap-around is eliminated). If you have one of the various add-on boards giving an 80-character screen line, or if you decide to off-print hard copy, you might want to modify these programs lines.

Lines 1750–1770 require the printing of B$, C$ and D$ if they are not null. "<>" is to be read as "is not equal to," and " " symbolizes the null string. B$, C$ and D$ have been created from S$ for the reason discussed in the previous paragraph.

Line 1780 inserts a blank line between sentences, then nulls B$, C$ and D$. Otherwise, if the next sentence generated has fewer than 40 characters, B$, C$ and D$ will "tag along" and be printed a second time. A$ is redefined automatically in line 1590 on each cycle.

Finally, although I have Dan Paymar's lowercase adapter installed in my Apple, I haven't yet figured out an easy way of getting uppercase and lowercase screen output for this program. It is possible to include a loop that reads each character in A$, except the first, and converts it to lowercase. But when I tried such a loop, it slowed down execution enormously, and wasn't really worth it.

Another possibility is to rewrite the program so that each time it is run it gets input for each category and converts the input to lowercase. Then a subroutine could be written to capitalize only the first letter of A$. But that would mean reentering each word each time the program is called up.

There are one or two other possibilities, such as using disk data files that are automatically read in when the program runs, but I haven't gotten around to that yet.

A Last, Mild Caution

While the sentences generated by the following program are not exactly X-rated, some have a certain risque quality. You might want to modify the vocabulary before showing the program off to any visiting elderly female relative.■

The largest selection of software from the world's largest software publisher.

Product List NO. 19½

Write for our catalog with full program descriptions and specifications.

DISK OPERATING SYSTEMS

CP/M CONFIGURED FOR:
- APPLE II
- DIGITAL MICROSYSTEMS FDC3
- DURANGO F-85
- HEATH H8 AND H89
- ICOM MICRO DISK
- ICOM 3712
- ICOM 3812
- ICOM 4511/PERTEC D3000
- INTEL MDS
- MICROPOLIS FLOPPY DISK
- MITS/ALTAIR
- MOSTEK MDX
- NORTH STAR
- OSI C3
- PRO-TECH HELIOS
- TRS-80 MODEL I
- TRS-80 MODEL II
- TRS-80 MODEL III
- ZENITH Z89

MP/M FOR INTEL MDS

HARD DISK INTEGRATION MODULES

- CORVUS WITH APPLE II SOFTCARD
- CORVUS WITH S100 AND TRS-80 MODEL II
- ICOM 4511/PERTEC D3000
- KONAN PLUS CDC PHOENIX
- XCOMP SM/S PLUS CDC PHOENIX
- XCOMP DFC10 FOR PERTEC D3000

SYSTEMS TOOLS

- BUG AND uBUG
- DESPOOL
- DISILOG
- DISTEL
- EDIT
- EDIT-80
- *FILETRAN*
- IBM/CPM
- MAC
- MACRO-80
- PASM
- PLINK
- RAID
- RECLAIM
- SID
- TRS-80 MODEL II CP/M CUSTOMIZATION DISK
- UNLOCK
- WORD-MASTER
- XASM-18
- XASM-48
- XASM-65
- XASM-68
- XMACRO-86
- ZDT
- Z80 DEVELOPMENT PACKAGE
- ZSID

TELECOMMUNICATIONS

- BSTAM
- BSTMS

LANGUAGES

- ALGOL-60
- APL/V80
- BASIC-80 (COMPILER)
- BASIC-80 (INTERPRETER)
- BDS C COMPILER
- CBASIC-2
- CIS COBOL
- COBOL-80
- FORTRAN-80
- KBASIC
- muLISP
- muSIMP
- NEVADA COBOL
- PASCAL/M
- PASCAL/MT
- PASCAL/MT +
- PASCAL/Z
- PL/I-80
- SMAL/80
- S-BASIC
- TINY C
- W'SMITHS C COMPILER
- XYBASIC

LANGUAGE AND APPLICATIONS TOOLS

- BASIC UTILITY DISK
- DATASTAR
- FABS
- FORMS 2 FOR CIS COBOL
- MAGSAM III
- MAGSAM IV
- MDBS.DRS
- MDBS.RTL
- M/SORT FOR COBOL-80
- PEARL
- QSORT
- STRING BIT
- STRING/80
- SUPER-SORT
- ULTRASORT II

WORD PROCESSING SYSTEMS AND AIDS

- MAGIC WAND
- LETTERIGHT
- MICROSPELL
- SPELLGUARD
- TEX
- TEXTWRITER III
- WORD-STAR
- WORDINDEX

DATA MANAGEMENT PACKAGES

- CONDOR
- HDBS
- MDBS
- MDBS.QRS
- WHATSIT?

GENERAL PURPOSE APPLICATIONS

- CBS
- SELECTOR III-C2
- SELECTOR IV

MAIL LIST APPLICATIONS

- MAILING ADDRESS (PTREE)
- MAIL-MERGE FOR WORD-STAR
- NAD
- POSTMASTER

BUSINESS APPLICATIONS

- ACCOUNTS PAYABLE (PTREE)
- ACCOUNTS PAYABLE (SSG)
- ACCOUNTS RECEIVABLE (PTREE)
- ACCOUNTS RECEIVABLE (SSG)
- GENERAL LEDGER II (CPAIDS)
- GENERAL LEDGER (PTREE)
- GENERAL LEDGER (SSG)
- GLECTOR FOR SELECTOR III-C2
- INVENTORY (PTREE)
- INVENTORY (SSG)
- PAYROLL (PTREE)
- PAYROLL (SSG)

WordStar users unite . . . with

WORDINDEX

Designed to assist users of WordStar maintain large documents, WORDINDEX is a list-making, tabulating and indexing champion.

Generate a table of contents, a list of figures and a list of tables, a sorted index, with index words on two levels: master and sub-references. In addition, you can re-number section numbers and page references, whenever the document has been changed. All automatically. **WORDINDEX** has the added attraction of user-selectable print control for all headings.

WORDINDEX is also useful for indexing publications not in machine-readable form.

WORDINDEX $195. Program With Full Documentation

MICROSPELL SPELLING CORRECTION PROGRAM AVAILABLE FOR $295.

NUMERICAL PROBLEM-SOLVING TOOLS

- ANALYST
- FPL
- muMATH
- STATPAK
- T/MAKER

OTHER SPECIALIZED APPLICATIONS

- DATEBOOK
- ESQ-1
- MASTER TAX
- PAS-3 DENTAL
- PAS-3 MEDICAL
- PROPERTY MANAGEMENT (PTREE)
- STANDARD TAX

BOOKS AND PERIODICALS

- APL—AN INTERACTIVE APPROACH
- ACCOUNTS PAYABLE & ACCOUNTS RECEIVABLE-CBASIC
- 8080/Z80 ASSEMBLY LANGUAGE
- THE CP/M HANDBOOK
- THE C PROGRAMMING LANGUAGE
- FIFTY BASIC EXERCISES
- GENERAL LEDGER-CBASIC
- LIFELINES
- PASCAL USER MANUAL AND REPORT
- PAYROLL WITH COST ACCOUNTING-CBASIC
- STRUCTURED MICROPROCESSOR PROGRAMMING
- USING CP/M—A SELF-TEACHING GUIDE

ACCESSORIES

- DC 300 DATA CARTRIDGE
- HEAD CLEANING DISKETTE
- FLIPPY DISK KIT
- FLOPPY SAVER

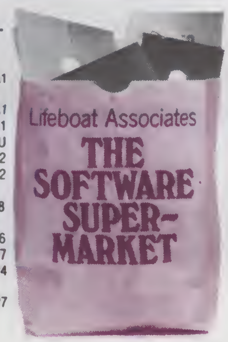
Program names and computer names are generally trademarks or service marks of the author or manufacturing company.

All software products have specific requirements for hardware and additional associated software (e.g. operating system or language).

All products are subject to terms and conditions of sale.

Ordering Information COMPUTERS SUPPORTED WITH MEDIA FORMAT ORDERING CODES.

ADDS Multivision	RT
AVL Eagle	RB
Altair 8800	B1
Altos	A1
Apple CP/M 13 Sector	RG
Apple CP/M 16 Sector	RR
BASF System 7100	RD
Blackhawk Micropolis Mod II	Q2
Blackhawk Single Density	Q3
CDS Versatile 3B	Q1
CDS Versatile 4	Q2
COMPAL-80	Q2
CSSN Backup	T1
Cromemco System 3	A1
Cromemco Z2D	R6
Delta Systems	A1
Digi-Log Microterm II	RD
Digital Microsystems	A1
Durango F-85	RL
Dynabyte DB8/2	R1
Dynabyte DB8/4	A1
Exidy Sorcerer + Lifeboat CP/M	Q2
Heath H8 + H17/H27 disk	P4
Heath H89 + Lifeboat CP/M	P4
Heath H89 + Magnolia CP/M	P7
Helios II	B2
ICOM 2411 Micro Floppy	R3
ICOM 3712	A1
ICOM 3812	A1
ICOM 4511 5440 Cartridge CP/M	D1
ICOM 4511 5440 Cartridge CP/M	D2
IMS 5000	RA
IMS 8000	A1
IMSAI VDP-40	R4
IMSAI VDP-42	R4
IMSAI VDP-44	R5
IMSAI VDP-80	A1
ISC Intecolor 8063/8360/8963	A1
Intertec Superbrain DOS 0.1	R7
Intertec Superbrain DOS 0.5	RJ
Intertec Superbrain DOS 3.x	RK
Intertec Superbrain QD	RS
Kontron PSI-80	RF
MITS 3200-3202	B1
MSD 5.25 in	RC
Meca Delta-1 5.25 in	P6
Micromation	A1
Micropolis Mod I	Q1
Micropolis Mod II	Q2
Morrow Discus	A1
Mostek	A1
North Star Single Density	P1
North Star Double Density	P2
North Star Quad Density	P3
Nylac Micropolis Mod II	Q2
Nylac Single Density	Q3
Ohio Scientific C3	A3
Onyx C8001	T2
Pertec PCC 2000	A1
Processor Technology Helios II	B2
Quay 500	RO
Quay 520	RP
RAIR Single Density	R9
RAIR Double Density	RE
Research Machines 5.25 in	RH
Research Machines 8 in	A1
Rex	Q3
SD Systems 5.25 in	R3
SD Systems 8 in	A1
Sanyo 7000 5.25 in	RQ
Spacebyte	A1
TEI 5.25 in	R3
TEI 8 in	A1
TRS-80 Model I STANDARD	R2
TRS-80 Model I + FEC Freedom	RN
TRS-80 Model I + Micromation	A4
TRS-80 Model I + Omikron 5.25 in	RM
TRS-80 Model I + Omikron 8 in	A1
TRS-80 Model I + Shuffleboard	A1
TRS-80 Model II	A1
TRS-80 Model III	RU
Vector MZ	Q2
Vector System B	Q2
Vista V-80 5.25 in Single Dens	R8
Vista V200 5.25 in Double Dens	P6
Zenith Z89 + Heath CP/M	P7
Zenith A89 + Lifeboat CP/M	P4
Zenith Z89 + Magnolia CP/M	P7



Lifeboat Associates, 1651 Third Avenue, N.Y., N.Y. 10028
(212) 860-0300 Telex: 640693 ✓360

In Germany,
Intersoft GmbH, Schlossgartenweg 5, D-8045 Ismaning **Telephone 089/966-444 Telex: 5213643 isof**

In Switzerland,
Lifeboat Associates GmbH, Aegeristr. 35, CH6340 Baar, **Telefon: 042/31 2931, Telex: 865265 MICO CH**

Now Your Child Can Learn More . . . At a Faster Pace

NEW!

GEOGRAPHY EXPLORER: U.S.A.

This program allows your child to travel the country and learn vital facts about each of our 50 states. Geography Explorer offers the most fascinating way of learning yet. Learn each state's name, capitol, largest city, nickname, etc. As a bonus, this package offers the capability of light pen use. **Order No. 0071RD $49.95**

NEW!

MATH MASTER

MATH MASTER is an educational package designed to help your child understand the mathematical concepts of addition, subtraction, multiplication and division through problem solving. This package also instructs the student in fraction problem solving. MATH MASTER will "walk" your child through the developmental concepts of mathematics with never-ending patience. **Order No. 0257R $9.95**

WORDWATCH

Wordwatch consists of 4 different programs, each designed to enhance your child's understanding of word relationships—this results in an improved vocabulary. In WORDRACE, the student must choose the proper definition for a given word. HIDE 'N SPELL asks the student to find misspellings. In SPELLING BEE, the student takes a pre-recorded quiz in which words are played aloud! Completing the package is SPELLING TUTOR, where words are jumbled, reversed or otherwise altered, and it's up to the student to straighten them out! Together, these four programs will help decrease dictionary dependency! **Order No. 0111R $7.95**

NEW!

OMNI-CALCULATOR

This package is a brilliant solution to the problems involving the complex world of different units of measure. It provides rapid means of conversion from one unit of measure to another in any of 10 categories: length, volume, mass, velocity, area density, power, energy, pressure/stress and temperature. Omni Calculator will give your child a great learning advantage, a head start in understanding the complex relationships between different units of measure. **Order No. 0211RD $19.95**

NEW!

THE ELEMENTS

This program can be used to introduce students to the periodic table of the elements, or for review by students and adults who want to refresh their memory in chemistry. It includes each of the elements' name, atomic number, weight and symbol, as well as its acid/base and normal physical state. Enrich your child's knowledge of science with the help of THE ELEMENTS. **Order No. 0216R $9.95**

LIFE

Even if you've only been involved with computers for a short while, you're sure to have heard of LIFE. This program is the computerized simulation of the life cycle of a colony of bacteria. It is based on a few simple concepts, but the results are captivating, animated graphics and an enlightening introduction into the world of biology and genetics. The LIFE program may be just the motivation your child needs to pursue an interest in the sciences. **Order No. 0078R $9.95**

TO ORDER: For the TRS-80*

See your local Instant Software dealer. If these packages are unavailable, order direct or

CALL TOLL FREE 1-800-258-5473

(We ship within 48 hours)

We Guarantee It!

Instant Software Guarantee

OUR PROGRAMS ARE GUARANTEED TO BE QUALITY PRODUCTS. IF NOT COMPLETELY SATISFIED YOU MAY RETURN THE PROGRAM WITHIN 60 DAYS. A CREDIT OR REPLACEMENT WILL BE WILLINGLY GIVEN FOR ANY REASON.

"The microcomputer will be a powerful tool in education. One of its greatest advantages is its interactive nature—the student becomes an active participant in learning, and the computer can give immediate feedback. The student does not have to wait until the lesson is complete or the teacher is free to check his work; the computer can evaluate the work immediately, and reward him if it is correct or explain why if the work is incorrect. The computer can then determine, on the basis of his work, whether the student is ready to go on to the next lesson or needs to review the present one.

"Educational computer programs will be available for students at all levels—from pre-school to post-college, from remedial to gifted—and in any subject. These programs can give a child a headstart in school; they can reinforce and provide practice in concepts from hand-eye coordination to Math to English or even Health. A computer can individualize a program to a child's level and pace. Programs will challenge the gifted child, allowing him to advance at his own pace. Best of all, educational computer programs can make learning fun."

Mary Shooshan, Educational Software Editor and former teacher

*TRS-80 is a trademark of the TANDY CORPORATION.

Instant Software™

All packages are on cassette except for Geography Explorer: USA, Omni-Calculator and Teacher's Aide which are on disk.

PETERBOROUGH, N.H. 03458
603-924-7296

At 5½, she's reading and thinking at a 4th grade level.

Is she a prodigy? Not at all! When learning is fun, your child can rapidly increase reading comprehension and reasoning power beyond current age level.

Now you can advance your child's educational level—without coaxing or hiring an expensive tutor. Everyone knows the best student-teacher ratio is 1:1. In modern classrooms, this is impossible. But in your own home, it's as easy as C.A.I. (Computer-Aided Instruction). Without the distractions of a crowded classroom, students can progress, without pressure, at their own pace.

Computers are fun, and once the fun is introduced, studying becomes a pleasure—instead of a chore. The ability to interact with the computer will motivate your child to learn much more than from a book that cannot respond—a book that is too easily closed. These Instant Software programs will guide and teach each lesson with the infinite patience only a computer can provide. Each of the twelve packages have been written and edited by professional educators who know what it takes to make a child want to learn. You can introduce your child to the world of learning by computer with Instant Software's Educational Software Series. All 12 packages are available now.

ARCHIMEDES' APPRENTICE

Archimedes Apprentice is a geometry package that will teach your child the formulas used to find the volume of any solid object, including parallelopipeds (cubes and rectangular solids), prisms, pyramids, cylinders, cones and spheres. It *will even quiz* the student on how well the lesson was learned. Give your child the advantage of learning these concepts at home, before being faced with them in a classroom. **Order No. 0092R $9.95**

TYPING TEACHER

A complete seven-part package that guides you from familiarization of the keyboard through typing words and phrases to mastery of touch-typing. Your video monitor becomes a bottomless page for typing practice, and your own private TYPING TEACHER, ready to teach when you're ready to learn. **Order No. 0099R $9.95**

TEACHER'S AIDE

Now you can have the benefit of Computer Assisted Instruction (CAI) in your own home. The Teacher's Aide program allows you to create tailor-made lessons for your child. The features of this program include the ability to review material *before taking* the lesson, a provision for hints to help answer questions and graphic displays as a reward for answering all questions correctly. Once you've created a lesson, you can save it and create an entire sequence of lessons.

The Teacher's Aide package is perfect for parents, teachers, and students who need the unlimited patience only a computer can provide. **Order No. 0214RD (disk-based) $39.95**

VIDEO SPEED READING TRAINER

With this package, students can increase their reading speed and comprehension. How? With practice! This three part program will flash characters, words and phrases onto the monitor screen. You must then type what you saw on the keyboard and enter it. You can begin at a relatively slow speed. As your speed and comprehension increase, the computer will automatically advance your speed. **Order No. 0100R $9.95**

IQ TEST

Are you smart enough to buy this package? IQ TEST will administer and score an intelligence test in 30 minutes. There are three equivalent tests, each consisting of 35 questions, designed to test your general knowledge and problem-solving abilities. Most of us claim a "touch of genius"—here's your chance to prove it! **Order No. 0157R $9.95**

VOCABULARY BUILDER

One of the most valuable tools your child can possess is a comprehensive vocabulary.With Instant Software's VOCABULARY BUILDER package, you can help your child develop a better understanding of the English language—and the fun is in the learning! The crossword puzzle format of these programs will hold the attention of the most reluctant student. The first program in this package is "Polonius," offering 140 crossword puzzles. The second is "VeeBeeGee," a program that provides the letters and allows the student to construct a puzzle on the game board. The VOCABULARY BUILDER package is educational entertainment for all ages—and it's never at a loss for words. **Order No. 0198R $9.95**

Instant Software™

Prices subject to change without notice.

PETERBOROUGH, N.H. 03458
603-924-7296

✓Reader Service—see page 210

The Many Moves of APF

By Dan Keen and Bob Kenney

Once you learn a little BASIC, it isn't too hard to figure out how to write a program to draw lines on the screen like an Etch-A-Sketch. But with the APF Imagination Machine, you can also make different shapes, in any of eight colors.

Furthermore, you're not limited to graphics. Any alphanumeric character can be printed and changed at any time during drawing, even in reverse video. It's also handy having the joysticks to control the cursor.

```
10 CALL 17046
20 DIM A$(10)
30 X=512: Y=129
35 POKE X,Y
50 A$=KEY$(2)
60 IF A$="N" THEN X=X-32: GOTO35
70 IF A$="S" THEN X=X+32: GOTO35
80 IF A$="E" THEN X=X+1: GOTO35
90 IF A$="W" THEN X=X-1: GOTO35
100 IF A$="2" THEN Y=Y-1: GOTO35
110 IF A$="1" THEN Y=Y+1: GOTO35
120 GOTO 50
```

Program listing.

In this program, a cursor, the shape and color of the block that will next be drawn, is displayed on the screen. When you hit the 1 key on the keypad of the left-hand controller, the cursor will change to the next ASCII character. Hitting the 2 will step back to the previous character. This way, you can scan through all the ASCII codes. Due to the machine's ability to handle ASCII numbers that are multiples of the 256 codes, no limit is needed in the program to check to see if the variable Y is less than zero or greater than 256. For example, if Y is 65, the letter A is displayed, as it is if Y equals 321 (65 + 256).

Movement is determined by moving the joystick east, west, north and south.

Since the computer supports wraparound, you don't need to include program lines to check the vertical or horizontal values to be plotted.

The short program given here is in an experimental form. This is an excellent way to learn about the graphics system in the APF.

Fig. 1 shows the 16 graphics shapes available and the relationship of their color to the ASCII codes. ASCII codes 128 to 143 give green shapes, 144 to 159 draw yellow ones and so on.

For example, a CHR$(143) displays a dark green box. A CHR$(143 + 16) is the same shape, only the color is yellow. A blue box is 143 + 32, and so on.

The video memory locations go from 512 to address 1023, as shown in Fig. 2.■

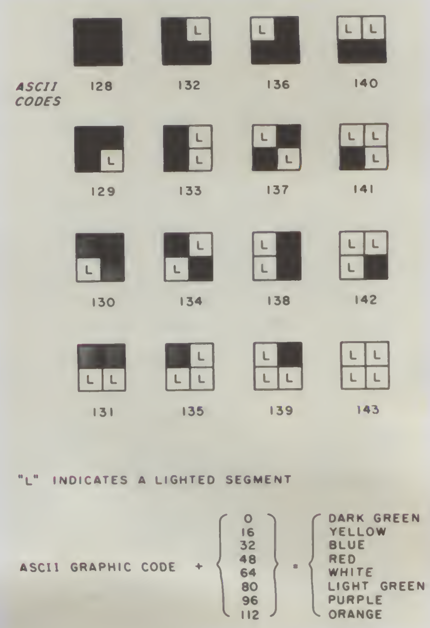


Fig. 1.

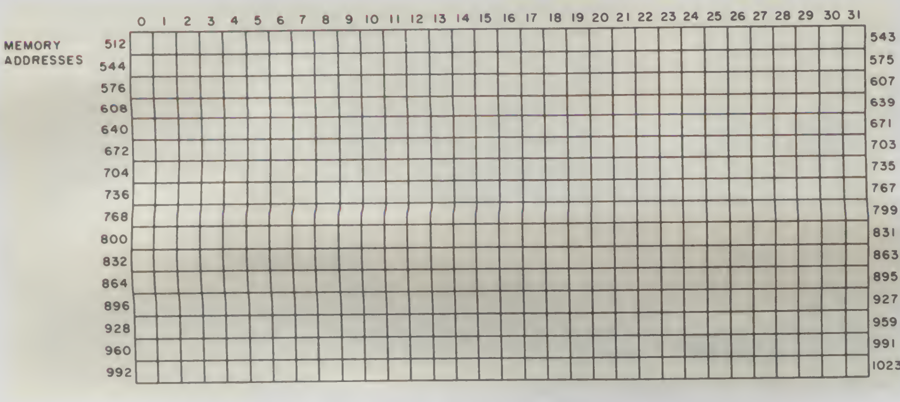
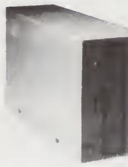


Fig. 2. Video display workshop.

Address correspondence to Dan Keen and Bob Kenney, RD 1, Box 432, State Highway 83, Cape May Court House, NJ 08210.

WE WILL NOT BE UNDERSOLD

DISK DRIVES



FOR TRS-80* Model I

CCI-100 TEAC	5¼", 40 Track (102K)	$314
CCI-100 MPI	5¼", 40 Track (102K)	$319
CCI-280	5¼", 80 Track (204K)	$429

ADD-ON DRIVES FOR ZENITH Z-89

CCI-189	5¼", 40 Track (102K)	$394
CCI-289	5¼", 80 Track (204K)	$499
Z-87	Dual 5¼" system	$995

External card edge and power supply included. 90 day warranty/one year on power supply.

RAW DRIVES	8" SHUGART 801R	$425
5¼" MPI, TEAC or TANDOM		$ CALL

MORROW DESIGNS/THINKER TOYS™

DISCUS 2D	1 DRIVE	$ 938	2 DRIVE	$1635
DISCUS 2+2	1 DRIVE	$1259	2 DRIVE	$2245
DISCUS Hard Disk	M26	$3990	M10	$2999

DEI CARTRIDGE TAPE BACK-UP

For your hard disk. With either S-100 control card or piggy back board for single board Z-80 computers. ~~$3500~~ $2995

DISK OPERATING SYSTEMS

PATCHPAK #4 by Percom Data				$8.95
CP/M*	for Model I, Zenith	$145	for Model II, Altos	$ 169
NEWDOS	Plus 40track	$ 79	NEWDOS 80	$ 135

DISKETTES — Box of 10 with plastic library case

5¼"	Scotch	$35	Maxell	$40	BASF/Verbatim	$24
8"	Scotch	$50	Maxell	$55	BASF/Verbatim	$36

CLEAR PLASTIC CASE–Holds 50 5¼" diskettes $19

COMPLETE SYSTEMS

ARCHIVES				$ CALL
ALTOS	ACS8000 Series			$ CALL
APPLE	II-16K	$1075	III-96K	$3749
	Call for other Apple products			
TRS-80*	II-64K	$3499	III-16K	$ 899
ZENITH	48K, all-in-one computer			$2395
ZENITH	Z-19			$ 795
TELEVIDEO	920C	$ 748	950	$1049
IBM	3101 Display Terminal			$1195
ATARI	400	$ 479	800	$ 795
APF	Game Only	$ 95	System	$ 489
MATTEL	INTELLIVISION			$ 229
Used TRS-80* Model I Computers, tested and guaranteed				$ CALL

MONITORS

APF	9" B & W TVM-10			$120
BELL & HOWELL	9" B & W BHD911			$220
LEEDEX	12" B & W	$129	13" Color	$389
SANYO	9" B & W VM4509			$155
SANYO	12" B & W DM5012			$226
SANYO	12" Green Screen DM5112			$238
SANYO	13" Color DMC6013			$416
ZENITH	13" Color			$350

TELECOMMUNICATIONS

LIVERMORE STAR MODEM 2-year guarantee	$129
UNIVERSAL DATA SYSTEMS UDS-103	$179
D-CAT HARD WIRED DIRECT MODEM	$189
AUTO-CAT Auto Answer, Direct Connect Modem	$249

PRINTERS

NEC Spinwriter

Letter Quality High Speed Printer

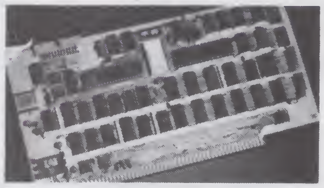
R.O.	$2395
R.O. with tractor feed	$2595
KSR with tractor feed	$2895

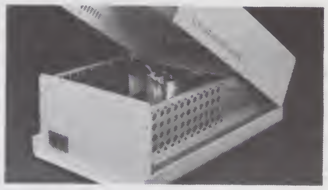
C.ITOH	Starwriter	$1795	Starwriter II	$1995
EPSON MX-80				$ CALL
PAPER TIGER				
IDS 445	Graphics & 2K buffer			$ 699
IDS 460	Graphics & 2k buffer			$1050
IDS 560	Graphics			$1599
ANADEX	DP-8000	$849	DP-9500/01	$1345
OKIDATA				
Microline 80	Friction & pin feed			$ 499
Microline 80	Friction, and pin & tractor feed			$ 599
Microline 82	Friction & pin feed feed			$ 719
Microline 83	120 cps, uses up to 15" paper			$ 995
CENTRONICS	730 $ 595	737 $ 780	779	$ 969
EATON	LRC 7000 +			$ 269
TI-810				
TRS-80* software, compressed print & vert. form control				$1865

16K RAM KITS 2 for $56 $30

200 ns for TRS-80*, Apple II, (specify): Jumpers $2.50

S-100 CALIFORNIA COMPUTER SYSTEMS





MAINFRAME	Model 2200A	$349
Z80 CPU	Model 2810	$269
MOTHER BOARD	Model 2501	$ CALL
16K STATIC RAM, 200ns	Model 2016BC	$309
32K STATIC RAM, 200ns	Model 2032C	$619
64K DYNAMIC RAM	Model 2065A	$599
FLOPPY DISC CONTROLLER	Model 2422A	$359
EXTENDER BOARD	Model 2520	$ CALL
2P + 2S I/O	Model 2718	$ CALL

ACCESSORIES

Z-80 SOFTCARD	$299.00
SCOTCH HEAD CLEANING DISKETTE: Cleans drive Read/Write head in 30 seconds; specify 5¼" or 8".	$ 25.00
FLOPPY SAVER: Protection for center holes of 5¼" floppy disks. Installation tools and rings for 25 diskettes.	$ 11.95
Re-orders of rings only	$ 6.95
VIDEX BOARD 80 Column, U/L case conversion card	$279.00
CRT FILM: Helps eliminate external glare, 9"	$ 29.00
RF MODULATOR: Adapts video to TV	$ 29.00
TRS-80 & OTHER MYSTERIES	$ 18.95
NEC SPINWRITER THIMBLE $11.95 **RIBBON**	$ 6.00
CCS CARDS: Parallel or serial printer interface cards	$115.00
RS232: For Radio Shack Interface.	$ 89.00
DISK-DRIVE EXTENDER CABLES: Fits all mini-disk drives.	$ 16.95
SIX (6) PRONG ISOLATOR: ISO-2	$ 54.00
AC FILTER/6 PRONG POWER STRIP	$ 39.00
DISK DRIVE CABLES: 2 drive $29.00 4 drive	$ 35.00

For fast delivery, send certified checks, money orders or call to arrange direct bank wire transfers. Personal or company checks require two to three weeks to clear. All prices are mail order only and are subject to change without notice. Call for shipping charges.

DEALER (NATIONAL/INTERNATIONAL) INQUIRIES INVITED **Send for FREE Catalogue**

The CPU SHOP ✓256

TO ORDER CALL TOLL FREE 1-800-343-6522
TWX: 710-348-1796 Massachusetts Residents call 617/242-3361

5 Dexter Row, Dept. K04M
Charlestown, Massachusetts 02129
Hours 10AM-6PM (EST) Mon.-Fri. (Sat. till 5)

Technical Information call 617/242-3361
Massachusetts Residents add 5% Sales Tax
Tandy Corporation Trademark/* Digital Research

✓Reader Service—see page 210

This SWTP flowchart generator program takes the tedium out of documenting programs.

Go with the Flow

By Peter A. Stark

Documenting programs so someone else—or even the programmer who wrote them—can understand them is not easy. It's one reason why there has been a big hullaballoo about structured programming and languages such as Pascal in recent years.

A common way of documenting programs is to draw a flowchart of them, but this is tedious, and many of us don't bother. It is possible, however, to have the computer help in the job. That is the purpose of FLOGEN, a BASIC program whose job is to draw a "flowchart" of other BASIC programs.

Obviously, a flowchart generator isn't intelligent enough to figure out what another program does. The best it can do is present the information that is already there. So the resulting flowchart will be similar to the program itself. Another limitation of a flowcharting program is that only a plotter can draw decent flowcharts; doing it on a printer is messy. Moreover, drawing a full flowchart on a printer can take ages.

To save time—and paper—FLOGEN doesn't print a full flowchart with squares, diamonds and all the other customary shapes. Instead, it prints a shortened version of the program, using only BASIC keywords such as PRINT, GOTO, IF or RESTORE, and uses arrows to indicate flow through the program.

Listing 1 is a good example of what FLOGEN does. The original program is shown in 1a; the processed flowchart is in 1b. Each BASIC statement is abbreviated to just the keyword. REMs usually don't make much sense if separated from the rest of the text, so they are deleted. They also clog up the flowchart; their removal opens up a space that separates sections of the program.

Every FOR-NEXT loop is shown at

Listing 1. Flowchart program example.

```
0010 PRINT "HELLO"
0020 REM DEMONSTRATION OF AUTOMATIC FLOWCHARTING
0030 FOR I=1 TO 15
0040 X=1
0050 FOR J=1 TO I
0060 X=X*J
0070 IF X>1E+06 GO TO 110
0080 NEXT J
0090 PRINT I, X
0100 NEXT I
0110 STOP
```

a. Sample program to be flowcharted.

```
                    0010 PRINT
                    0020
!-------------------0030 FOR I
!                   0040  X=
! !-----------------0050 FOR J
! !                 0060  X=
! !                 0070 IF > > > > > > > > > > > > > > > > > > > > >!
! ------------------0080 NEXT J                                       !
!                   0090 PRINT                                        !
--------------------0100 NEXT I                                       !
                    0110 ***STOP*** < < < < < < < < < < < < < < < < <<
```

b. Resulting flowchart.

Listing 2. An example of a complex flowchart.

```
0010 PRINT "THIS IS A TEST PROGRAM"
0020 GOTO 60
0030 PRINT J
0040 NEXT J
0050 GOTO 80
0060 FOR J=1 TO 5
0070 GOTO 30
0090 END
```

a. Sample program.

More →

Address correspondence to Peter A. Stark, PO Box 209, Mt. Kisco, NY 10549.

the left, and it's easy to spot loops that are not properly nested because the loop markers will cross over. FOR and NEXT statements also are printed with the loop variable, so they are easier to follow.

Assignment statements such as X=1 are shortened to stop at the equal signs. If several statements appear on one line, each is shortened.

Transfers of control with GOTO or IF-THEN statements are shown on the right of the flowchart with arrows. The symbols used are < for left, > for right, ! for down and ↑ for up. When both < and > are needed on the same line, FLOGEN, uses - - -. In this example, it's clear that the IF in line 70 sends the program to line 110.

In simple programs such as this, the transfers of control are easy to follow. When many jumps occur in the same program, the picture becomes a little harder to follow. Listing 2a shows a program that deliberately has some bad programming practices. It's written so that the NEXT I appears before the FOR I= statement, and it has GOTOs all over the place. It even has a GOTO 80, without a line 80. FLOGEN has printed error messages pointing to the FOR-NEXT problem and showing that line 80 is missing.

It takes a while to get accustomed to interpreting the arrows on the right side of the flowchart. Follow each path through from its start to the end. Probably the hardest one to see is 70 GOTO 30. As you follow the >>> arrows right from the 0070 GOTO, don't become sidetracked when you reach the >! and mistakenly go down. Note that right behind the >! is another >, so keep going until you reach the ↑ and there is no more. As you study this sample, you will see careful organization to the arrows—horizontal arrows are only on odd locations, while vertical arrows are only on even locations. When you get to any arrow, take it only if there is no continuing arrow just past it.

For instance, statement 70 GOTO 30 looks like this:

0070 GOTO >>>>>>>>>>!>↑

As you go right, following the >>>, you come to a !, but there is another > just after it. So you keep going until you get to the ↑ and follow it up to a << at line 30. You must take that turn because there is no other ↑ above that.

The FLOGEN Program

FLOGEN appears in Listing 3. If you want to customize it, the variable

Listing 2 continued.

```
                    0010 PRINT
                    0020 GOTO > > > > > > > > > > > > > > > > > > > >!
                    0030 PRINT  < < < < < < < < < < < < < < < < < <<  !
ERROR? - NEXT WITHOUT FOR?
                    0040 NEXT J                                       !
                    0050 GOTO > > > > > > > > > > > > > > > > > >!  ^ !
!-------------------0060 FOR J< < < < < < < < < < < < < < < < < <!<^<<
!                   0070 GOTO > > > > > > > > > > > > > > > > > >!^^
!                   0090 ***END***                                 !
ERROR - MISSING NEXT J
ERROR - MISSING LINE NO. 80
```

b. Flowchart.

Listing 3. FLOGEN program.

```
0005 REM VERSION 1.0PD

0010 PRINT "FLOGEN - BASIC FLOWCHART PROGRAM"
0020 PRINT
0030 PRINT
0040 PRINT : PRINT
0050 STRING=80
0060 INPUT "ENTER FILE NAME OF SOURCE PROGRAM", N$
0070 INPUT "WHAT DRIVE IS IT IN", D
0080 IF D=1 GO TO 0110
0090 IF D=2 THEN N$="2/"+N$ : GO TO 0110
0100 GOTO 070                                    :REM IF INVALID DRIVE NUMBER
0110 DIM S(250), S1(250) : S6=0 : S5=1           :REM X-REFERENCES
0120 K9=32                                       :REM 32 KEYWORDS
0130 DIM K$(K9)                                  :REM KEYWORDS
0140 DIM L$(10), R(20), H$(20)                   :REM LEFT, RIGHT, HORIZ
0150 LINE=0
0160 INPUT "PRINT ON WHAT PORT", P4
0170 GOSUB 3330                                  :REM GO READ KEYWORDS

0180 REM LOOP TO SCAN THROUGH SOURCE PROGRAM AND COMPILE A
0190 REM TABLE OF LINE NUMBER CROSS REFERENCES

0200 OPEN #11, N$
0210 LINPUT #11, A$                              :REM READ STATEMENT FROM DISK
0220 IF EOF(11)=1 GO TO 750                      :REM IF END OF FILE FINISH
0230 L=LEN(A$)
0240 IF L<7 THEN PRINT "LINE TOO SHORT: ";A$ : STOP
0250 S$=LEFT$(A$,6)                              :REM LINE NUMBER,SPACE, AND FIRST CHAR

0260 REM SKIP REM STATEMENTS

0270 IF MID$(A$,6,3)="REM" GO TO 210
0280 REM DELETE SPACES FROM LINE

0290 FOR I=7 TO L
0300     B$=MID$(A$,I,1)
0310     IF B$<>" " THEN S$=S$+B$
0320 NEXT I

0330 REM NOW SCAN LINE FOR OCCURRENCES OF
0340 REM GOTO, THEN

0350 L=LEN(S$)
0360 IF L<10 GO TO 210                           :REM CAN'T BE A TRANSFER
0370 FOR I=6 TO L
0380     B$=MID$(S$,I,4)
0390     IF B$=":REM" GO TO 210                  :REM SKIP REST OF REMARK
0400     IF B$="GOTO" GO TO 470
0410     IF B$="THEN" GO TO 470
0420     GOTO 440                                :REM NOTHING FOUND

0430     REM END OF LOOP TO SEARCH ONE LINE

0440 NEXT I
0450 GOTO 210

0460 REM NOW PROCESS LINE NUMBER

0470 B=4

0480 REM FOUND GOTO OR THEN
0490 REM TARGET LINE NUMBER. I POINTS TO ITS BEGINNING
0500 REM AND B IS ITS LENGTH

0510 C=VAL(LEFT$(S$,4))                          :REM CURRENT LINE NUMBER
0520 P=I+B                                       :REM DIGIT POSITION
0530 T$=MID$(S$,P,1)                             :REM DO A QUICK TEST
0540 GOSUB 3040                                  :REM TEST FOR A DIGIT
0550 IF T=0 THEN I=I+B : GO TO 440               :REM NEXT MUST BE DIGIT

0560 REM FOUND FIRST DIGIT OF TARGET LINE NO.
```

More

Listing 3 continued.

```
0570 D=VAL(T$)                                  :REM DIGIT
0580 FOR J=P+1 TO P+3                           :REM LOOK AT NEXT 3 CHARS
0590      T$=MID$(S$,J,1)
0600      GOSUB 3040
0610      IF T=0 GO TO 650                      :REM GOT IT ALL
0620      D=D*10+VAL(T$)                        :REM NEXT DIGIT
0630 NEXT J

0640 REM GOT TARGET LINE NUMBER IN D; NOW STORE IT
0650 S6=S6+1                                    :REM INCREMENT POINTER
0660 IF C<D THEN S(S6)=C+D/10000 : S1(S6)=1     :REM DOWN REF
0670 IF C>D THEN S(S6)=D+C/10000 : S1(S6)=2     :REM UPWREF

0680 REM CHECK FOR A SERIES OF LINE NUMBERS

0690 IF MID$(S$,J,1)="," THEN P=J+1 : GO TO 530

0700 REM DONE WITH GOTO,THEN,GOSUB,RESTORE

0710 GOTO 440

0720 REM ENTIRE PROGRAM HAS BEEN SCANNED AND S() HAS S6
0730 REM ENTRIES OF LINE NUMBERS.
0740 REM ON EOF,COME HERE
0750 REM NOW DO A SORT OF S() AND S1()

0760 M=S6
0770 M=INT(M/2)
0780 IF M=0 GOTO 960
0790 K=S6-M
0800 J=1
0810 I=J
0820 L1=I+M
0830 IF S(I)<S(L1) GOTO 920
0840 A=S(I)
0850 S(I)=S(L1)
0860 S(L1)=A
0870 A=S1(I)
0880 S1(I)=S1(L1)
0890 S1(L1)=A
0900 I=I-M
0910 IF I>=1 GOTO 820
0920 J=J+1
0930 IF J>K GOTO 770
0940 GOTO 810

0950 REM SORTING FINISHED. NOW START OUTPUT PASS.

0960 RESTORE #11
0970 LINPUT #11,S$                              :REM READ NEXT SOURCE LINE
0980 IF EOF(11)=1 GO TO 2440
0990 C$=LEFT$(S$,5)                             :REM LINE NUMBER
1000 B=6                                        :REM NEXT CHARACTER
1010 R4=0 : R5=0                                :REM ERASE DIRECTION POINTERS
1020 GOSUB 2570                                 :REM SEARCH FOR NEXT KEYWORD

1030 REM FIRST KEYWORD IS NOW FOUND. IF FOR OR NEXT
1040 REM UPDATE LEFT MARKERS

1050 L3=0                                       :REM LEFTMOST POSITION
1060 GOSUB 3110                                 :REM UPDATE LEFT MARKERS

1070 REM NOW APPEND KEYWORD TO LINE NUMBER

1080 IF K$="STOP" THEN K$="***STOP***"
1090 IF K$="END" THEN K$="***END***"
1100 IF K$="GOSUB" THEN K$="GOSUB-->"
1110 IF K$<>"REM" THEN C$=C$+K$+" "+V$

1120 REM LEFT MARKERS NOW UPDATED. NEXT LOOK FOR MORE
1130 REM KEYWORDS AND REPEAT,IF NECESSARY

1140 B=B+1
1150 A$=MID$(S$,B,1)
1160 IF A$="" GO TO 1320                        :REM FINISHED SEARCHING
1170 IF A$<>":" GO TO 1140
1180 B=B+1
1190 A$=MID$(S$,B,1)
1200 IF A$=" " GO TO 1180
1210 IF A$="" GO TO 1320                        :REM FINISHED SEARCHING
1220 GOSUB 2570                                 :REM SEARCH FOR NEXT KEYWORD
1230 GOSUB 3110                                 :REM UPDATE LEFT MARKERS
1240 IF K$="STOP" THEN K$="***STOP***"
1250 IF K$="END" THEN K$="***END***"
1260 IF K$="GOSUB" THEN K$="GOSUB-->"
1270 IF K$<>"REM" THEN C$=C$+" : "+K$+" "+V$    :REM SAVE KEYWORD
1280 GOTO 1140

1290 REM LEFT MARKERS AND KEYWORDS ARE TAKEN CARE OF.
1300 REM NOW DO THE RIGHT MARKERS.
1310 REM FIRST,RELEASE A COLUMN WHICH IS PARTIALLY RELEASED

1320 FOR K=1 TO 10
1330      IF R(K)=.5 THEN R(K)=0
1340      IF R(K)=.4 THEN R(K)=0
1350      IF R(K)=1 THEN R(K)=.5
1360      IF R(K)=.9 THEN R(K)=.4
```

More

list in Table 1 will help keep track of which variables are used where.

This version of FLOGEN is written for Percom Super BASIC and runs on an SWTP 6800 system using the Percom floppy-disk system. It can easily be modified to run on 8080, Z-80 or other systems using other microprocessor BASICs.

The important thing is to be able to read BASIC source programs as data files. Almost all disk systems can do this. With little work—reformatting a program tape to be acceptable as data—FLOGEN could also run on a cassette-based system, but the running time would probably be long. This is because FLOGEN reads the source program twice, one line at a time. It takes different amounts of time to read each line, depending on the required processing time, and it would be hard to start and stop the tape just right.

FLOGEN assumes that each line of source code contains a four-digit line number, padded with left-hand zeros if necessary. If your source files are stored in a slightly different way, some modification may be needed.

The second requirement is to have long enough string lengths that an entire line of source code will fit into one string. In this program, the string length is set to 80 characters, with STRING=80 in line 50. As written, FLOGEN also requires that numeric variables be floating point (not integer) and have at least eight significant digits.

Another requirement is to read an entire line of source code at a time, disregarding any commas, quotes, semicolons or other special delimiters. In Percom Super BASIC a special LINPUT statement exists just for that purpose; the new TSC 6800 BASIC has an INPUT LINE statement that does the same. Without this, the source statement

```
0010 REM AFTER THIS, MULTIPLY THE
     NUMBERS
```

is read only up to the comma delimiter and appears as

```
0100 REM AFTER THIS
```

In FLOGEN, the LINPUT is used in two places, lines 210 and 970, corresponding to the two passes through the source program. If your BASIC does not have such a special statement, a similar function can be built up out of the following sequence:

```
INPUT A$,W$,X$,Y$,Z$
IF W$<>"" THEN A$=A$+","+W$
IF X$<>"" THEN A$=A$+","+X$
IF Y$<>"" THEN A$=A$+","+Y$
IF Z$<>"" THEN A$=A$+","+Z$
```

Listing 3 continued.

```
1370 NEXT K

1380 REM IS THIS LINE THE END OF A TRANSFER FROM/TO ABOVE?

1390 L6=VAL(LEFT$(S$,4))                          :REM LINE NUMBER
1400 FOR K=1 TO 10
1410      IF ABS(R(K))<>L6 GO TO 1510

1420      REM THE END OF A TRANSFER. WAS IT FROM OR TO?

1430      IF R(K)>0 GO TO 1490

1440      REM R(K) IS NEGATIVE,SO IT WAS FROM (UP)

1450      IF R5<K THEN R5=K                       :REM UPDATE ARROW >
1460      R(K)=.9                                 :REM PARTIALLY RELEASE COLUMN
1470      GOTO 1510

1480      REM R(K) IS POSITIVE,SO IT WAS TO (DOWN)

1490      IF R4<K THEN R4=K                       :REM UPDATE ARROW <
1500      R(K)=1                                  :REM PARTIALLY RELEASE COLUMN
1510 NEXT K

1520 REM NEXT,IS THIS LINE THE BEGINNING OF A TRANSFER
1530 REM TO/FROM BELOW? CHECK LINE NO. TABLE

1540 IF S5=S6 GO TO 1600                          :REM DON'T INCREMENT PAST END
1550 T=INT(S(S5))
1560 IF T>=L6 GO TO 1600
1570 IF T<>0 THEN PRINT "ERROR - MISSING LINE NUMBER "; T
1580 S5=S5+1
1590 GOTO 1540
1600 IF INT(S(S5))<>L6 GO TO 1940                 :REM NO TRANSFER

1610 REM THIS LINE IS THE BEGINNING OF A TRANSFER

1620 D9=(S(S5)-INT(S(S5)))*10000                  :REM DESTINATION
1630 S(S5)=0                                      :REM ERASE IT

1640 REM CAN WE TIE INTO AN EXISTING COLUMN?

1650 FOR K=1 TO 10
1660      IF ABS(INT(R(K)))=D9 GO TO 1700
1670 NEXT K
1680 GOTO 1730                                    :REM NOTHING MATCHES,CAN'T TIE IN

1690 REM SAME DESTINATION - IS IT THE SAME DIRECTION?

1700 IF S1(S5)=1 THEN IF R(K)>0 GO TO 1830        :REM YES !
1710 IF S1(S5)=2 THEN IF R(K)<0 GO TO 1870        :REM YES ^

1720 REM NEED NEW COLUMN. FIND AN EMPTY ONE

1730 FOR K=10 TO 1 STEP -1
1740      IF R(K)=0 GO TO 1780
1750 NEXT K
1760 GOTO 1890                                    :REM ALL COLS BUSY,SO IGNORE

1770 REM FOUND AN EMPTY COLUMN. CHECK DIRECTION AND USE IT

1780 IF S1(S5)=1 GO TO 1820                       :REM DOWN
1790 IF S1(S5)=2 GO TO 1860                       :REM UP
1800 PRINT "ERROR ON "; S$ : STOP

1810 REM DOWN DIRECTION

1820 R(K)=D9
1830 IF R5<K THEN R5=K
1840 GOTO 1890

1850 REM UP DIRECTION

1860 R(K)=-D9+.5                                  :REM ADD TEMPORARY MARKER
1870 IF R4<K THEN R4=K

1880 REM NOW CHECK IF THERE ARE MORE TRANSFERS BELOW

1890 IF S5=S6 GO TO 1940                          :REM NO MORE
1900 IF INT(S(S5+1))=L6 THEN S5=S5+1 : GO TO 1540

1910 REM NO MORE TRANSFERS BELOW. LEFT AND RIGHT MARKERS ARE
1920 REM NOW PDATED,SO PRINT ENTIRE LINE.
1930 REM FIRST,DO LEFT (FOR-NEXT) MARKERS.

1940 FOR K=1 TO 10
1950      H$(K)="-"                               :REM HORIZONTAL LINE
1960      IF K>L3 THEN H$(K)=" "
1970 NEXT K

1980 REM NOW DO LEFT VERTICALS ETC

1990 FOR K=10 TO 1 STEP -1
2000      IF L$(K)="" THEN PRINT #P4,H$(K);H$(K);
2010      IF L$(K)<>"" THEN PRINT #P4,"!";H$(K);
2020 NEXT K
```

More

This sequence can take care of up to four commas or other delimiters in one line of source code. For example, if it reads the source code line

0010 REM AFTER THIS, MULTIPLY THE NUMBERS

the 0010 REM AFTER THIS reads into A$, and MULTIPLY THE NUMBERS goes into W$; X$, Y$ and Z$ are empty. Since W$ is not the empty string "" it is added to the end of A$ with a comma inserted between W$ and A$.

This is an effective method of dealing with the problem, although it is not foolproof. In some BASICs, portions of text enclosed in quotes are deleted, or other control characters may interfere with operation. (Also, any line ending with a comma will have that comma removed, but that shouldn't happen too often.) In this case, it may be necessary to experiment with your BASIC to find the instruction that tests for the quote or other character, and temporarily POKE a different character into it to change the test.

Other comments on specific statements that may not be the same in your BASIC:

Lines 60–100 enter an input file name. Percom Super BASIC needs a 2/ symbol before a file name to identify programs in drive two, so line 90 precedes the file name N$ with 2/ if the drive number D entered is 2.

Line 200 opens the input file, line 210 reads one line and line 220 tests for end of file. When the program tries to read the next line after the last, line 220 sends the program to line 750.

The MID$(A$,S,L) function returns a substring of length L, starting at the Sth character of A$. LEFT$(A$,L) returns the first L characters of A$.

How It Works

Lines 10–160 initialize variables.

Line 0170 goes to a subroutine to read K9 keywords from DATA statements. Typical keywords are DIM, REM, LET and any other word that defines a type of BASIC statement.

Line 200 opens the file source program file.

Line 210 starts the main loop, and reads the next line of source program.

Lines 230–240 quickly check the length of this line. Any line having fewer than seven characters cannot be a valid statement.

Line 250 gets the first six characters of the line; this consists of a four-digit

line number, a space and the first character of the line.

Line 270 skips all REM lines.

Lines 290–320 scan through the rest of the source line in A$, and remove all spaces.

Lines 350–450 now scan through the resulting string, looking for :REM, and for GOTO and THEN. If neither is found, the program goes back to line 210 to read the next line.

If a GOTO or THEN is found, lines 470–550 look at the next few characters to see whether a line number follows. (A THEN could be followed by another statement, instead of a line number.) If not, line 550 sends the program back to look for more. The I=I+B in line 550 changes the loop counter in the FOR-NEXT loop. It's not good programming practice, but it does save a little time. If your BASIC interpreter objects, just delete this portion of the line.

Once a line number is found, lines 570–630 separate the complete line number from the rest of the line. This line number becomes the destination of a transfer from the current line C, obtained in line 510, to the destination line D, obtained in line 620. Lines 660–670 check whether this is a transfer forward or backward in the program, and store both line numbers into one position of array S( ). The lower of the two is stored in the integer portion, while the higher line number is stored as a fraction.

For instance, a GOTO from line 5678 to line 1234 is stored in S( ) as 1234.5678, which explains why your BASIC must have at least eight-digit floating-point numbers. If you have an integer BASIC or one with fewer than eight digits, you have to store the two line numbers separately in two arrays, perhaps called S( ) and T( ).

At the same time the line numbers are stored in S( ), the corresponding location in array S1( ) holds a 1 if this is a transfer forward (down) in the program, or a 2 if it is a transfer backward (up).

Since ON . . . GOTO statements can have a series of line numbers separated by commas, line 690 checks for a comma after the line number and goes back to look for more numbers if a comma is found.

Line 710 ends the first pass. At the end of the pass, arrays S( ) and S1( ) hold a cross-reference table of the entire program, listing every transfer within the program except for GOSUBs and RETURNS.

Lines 760–940 now do a Shell sort

Listing 3 continued.

```
2030 REM NOW PRINT THE KEYWORDS

2040 PRINT #P4,C$;

2050 REM NOW DO THE RIGHT MARKERS.
2060 REM FIRST,FIGURE OUT THE HORIZONTAL LINE

2070 FOR K=1 TO 10
2080     H$(K)=" "
2090     IF K<=R4 GO TO 2150                    :REM IF LEFT

2100     REM NO LEFT

2110     IF K>R5 GO TO 2190                     :REM NO LEFT AND ALSO NO RIGHT

2120     REM RIGHT,NO LEFT

2130     H$(K)=">"
2140     GOTO 2190
2150     IF K>R5 GO TO 2180                     :REM IF LEFT,NO RIGHT
2160     H$(K)="-"                              :REM BOTH RIGHT AND LEFT
2170     GOTO 2190
2180     H$(K)="<"                              :REM LEFT,NO RIGHT
2190 NEXT K

2200 REM NOW SPACE OVER TO THE FIRST COLUMN OF RIGHT MARKERS

2210 IF POS=INT(POS/2)*2 THEN PRINT #P4, " ";
2220 P3=POS
2230 FOR K=P3 TO 49 STEP 2
2240     PRINT #P4,H$(1);" ";
2250 NEXT K

2260 REM NOW PRINT RIGHT VERTICALS ETC.

2270 FOR K=1 TO 10
2280     PRINT #P4,H$(K);
2290     IF R(K)>1 THEN PRINT #P4,"!";
2300     IF R(K)=1 THEN PRINT #P4,"<";
2310     IF R(K)=.9 THEN PRINT #P4,"^";
2320     IF R(K)=.5 THEN PRINT #P4," ";
2330     IF R(K)=.4 THEN PRINT #P4," ";
2340     IF R(K)=0 THEN PRINT #P4," ";
2350     IF R(K)>=0 GO TO 2390                  :REM IF NOT UPWARD
2360     IF R(K)=INT(R(K)) THEN PRINT #P4,"^"; : GO TO 2390
2370     PRINT #P4, "<";                        : REM TOP OF UPWARD TRANSFER
2380     R(K)=R(K)-.5
2390 NEXT K
2400 PRINT #P4

2410 REM ALL DONE PRINTING,NOW DO NEXT LINE

2420 GOTO 970

2430 REM END OF PRINTING; FINISH UP

2440 CLOSE #11

2450 REM CHECK IF THERE ARE ANY LEFTOVER MARKERS

2460 FOR K=1 TO 10
2470     IF L$(K)<>"" THEN PRINT "ERROR - MISSING NEXT ";L$(K)
2480 NEXT K
2490 FOR K=1 TO 10
2500     IF R(K)>1 GO TO 2530
2510     IF R(K)<0 GO TO 2530
2520     GOTO 2540
2530     PRINT #P4, "ERROR - MISSING LINE NO. ";INT(R(K))
2540 NEXT K
2550 END

2560 REM SUBROUTINE TO SEARCH LINE FOR NEXT KEYWORD

2570 FOR K=1 TO K9
2580     L=LEN(K$(K))
2590     IF MID$(S$,B,L)=K$(K) GO TO 2750
2600 NEXT K

2610 REM DIDN'T FIND,SEE IF IT'S AN IMPLIED LET

2620 FOR K=B TO 90
2630     T$=MID$(S$,K,1)
2640     IF T$="" GO TO 2730
2650     IF T$=":" GO TO 2730
2660     IF T$="=" GO TO 2700
2670 NEXT K
2680 PRINT "ERROR - UNDECIPHERABLE STATEMENT" : STOP

2690 REM FOUND AN =,ASSUME A LET

2700 T$=MID$(S$,B,K-B+1) : K=0
2710 GOTO 2750

2720 REM DIDN'T FIND ANYTHING; IS IT AN ERROR?

2730 K=K9
```

More

Listing 3 continued.

```
2740 GOTO 2750
2750 K$=T$
2760 IF K<>0 THEN K$=K$(K)
2770 V$=""

2780 REM FOR A FOR OR NEXT,GET THE VARIABLE TOO

2790 IF K$="FOR" GO TO 2830
2800 IF K$="NEXT" GO TO 2950
2810 RETURN

2820 REM AFTER A FOR,GET EVERYTHING TO =

2830 FOR K=B+3 TO 80
2840      IF MID$(S$,K,1)="=" GO TO 2880
2850 NEXT K
2860 PRINT "ERROR IN LINE ";S$ : STOP

2870 REM FOUND AN = AT POSITION K

2880 A$=MID$(S$,B+3,K-B-3)                          :REM VARIABLE

2890 REM NOW DELETE SPACES FROM IT

2900 FOR K=1 TO LEN(A$)
2910      IF MID$(A$,K,1)<>" " THEN V$=V$+MID$(A$,K,1)
2920 NEXT K
2930 RETURN

2940 REM AFTER NEXT,GET EVERYTHING TO STATEMENT DELIMITER

2950 FOR K=B+4 TO 80
2960      IF MID$(S$,K,1)="" GO TO 3010
2970      IF MID$(S$,K,1)=":" GO TO 3010
2980 NEXT K
2990 PRINT "ERROR IN LINE "; S$ : STOP

3000 REM FOUND DELIMITER AT POSITION K

3010 A$=MID$(S$,B+4,K-B-4)                          :REM VARIABLE
3020 GOTO 2900

3030 REM SUBROUTINE TO TEST CHARACTER T$ FOR A DIGIT
3040 REM T=1 FOR DIGIT,T=0 FOR NON-DIGIT

3050 T=0
3060 IF T$<"0" THEN RETURN
3070 IF T$>"9" THEN RETURN
3080 T=1
3090 RETURN

3100 REM SUBROUTINE TO UPDATE LEFT MARKERS

3110 IF K$="NEXT" GO TO 3240
3120 IF K$<>"FOR" THEN RETURN

3130 REM IT'S A FOR. FIND THE LEFTMOST EMPTY COLUMN FOR IT
3140 REM AND PLACE A MARKER FOR IT

3150 FOR K=10 TO 1 STEP -1
3160      IF L$(K)="" GO TO 3200
3170 NEXT K
3180 PRINT "ERROR - TOO MANY NESTED LOOPS IN "; S$ : STOP

3190 REM FOUND AN EMPTY COLUMN. ASSIGN IT TO A VARIABLE

3200 L$(K)=V$
3210 IF L3<K THEN L3=K
3220 RETURN

3230 REM IT'S A NEXT. DELETE LEFT MARKER

3240 FOR K=1 TO 10
3250      IF L$(K)=V$ GO TO 3290
3260 NEXT K
3270 PRINT "ERROR? - NEXT WITHOUT FOR?"
3280 RETURN
3290 IF L3<K THEN L3=K
3300 L$(K)=""                                       :REM ERASE MARKER
3310 RETURN

3320 REM SUBROUTINE TO READ K9 KEYWORDS

3330 FOR K=1 TO K9
3340      READ K$(K)
3350 NEXT K
3360 RETURN
3370 DATA DIM,DEF,FOR,NEXT,IF,RETURN,READ,INPUT
3380 DATA LINPUT,PRINT,OPEN,CLOSE,RESTORE,GOTO
3390 DATA GOSUB,ON,REM,GO TO,END,DATA,STOP
3400 DATA CHAIN,DIGITS=,LET,LINE=,POKE,PORT=
3410 DATA SCRATCH,STRING=,TRACE,FILESZ=
3420 DATA ***
```

of S( ) and sort S1( ) into the same order as S( ). This table is now used by pass two to plot transfers on the right of the flowchart.

To see what this transfer table does, look at the following program

```
10 X=1
20 GO TO 30
30 GO TO 10
```

After pass one, but before the table is sorted, S( ) and S1( ) have the following two entries each:

S(1) = 20.0030   S1(1) = 1 (down)
S(2) = 10.0030   S1(2) = 2 (up)

After S( ) is sorted, the two arrays now have:

S(1) = 10.0030   S1(1) = 2 (up)
S(2) = 20.0030   S1(2) = 1 (down)

These tables later tell FLOGEN that there should be a connection from line 10 to line 30 with arrows up, and a connection from line 20 to line 30 with arrows down.

Pass two starts in line 960 when the file is restored to the beginning. Lines 970–980 then read each line, starting from the beginning, and check for end of file.

Line 990 gets the first five characters of the line, which is the line number and a space. B in line 1000 then points to the next character in the line.

Line 1010 now initializes two variables called R4 and R5. These are used when < or > is printed on the right-hand side of transfers to indicate how far the transfers should go. Figure 1 shows how a line of the flowchart can be considered as divided into FOR-NEXT columns L10 through L1 on the left, and transfer columns R1 through R10 on the right. This line shows that we need left arrows out of column R7; variable R4 holds the column number to which we need to extend left arrows <; variable R5 holds the column to which we need right arrows >. In this case, R4 equals 7 and R5 equals 0 since no right arrows are needed.

Line 1020 goes to a subroutine that searches the source line for keywords (it compares the keywords read out of the DATA statements against the line).

Lines 1050 and 1060 now go to work on FOR-NEXT statements via a subroutine starting at line 3100. Whenever a FOR statement is found, the subroutine finds the leftmost empty FOR-NEXT column, starting from L10, and places a marker in it. These markers are held in array L$( ) and consist of the loop variable.

In the example of Fig. 1, lines 3150 through 3180 search through L$( ) for

```
NAME    USED IN LINES...
A       840  860  870  890
A$      210  230  240  250  270  300  1150 1160 1170 1190 1200 1210 2880 2900 2910
        2910 3010
B       470  520  550  1000 1140 1140 1150 1180 1180 1190 2590 2620 2700 2700 2830
        2880 2880 2950 3010 3010
B$      300  310  310  380  390  400  410
C       510  660  660  670  670
C$      990  1110 1110 1270 1270 2040
D       70   80   90   570  620  620  660  660  670  670
D9      1620 1660 1820 1860
H$()    140  1950 1960 2000 2000 2010 2080 2130 2160 2180 2240 2280
I       290  300  320  370  380  440  520  550  550  810  820  830  840  850  870
        880  900  900  910
J       580  590  630  690  690  800  810  920  920  930
K       790  930  1320 1330 1330 1340 1340 1350 1350 1360 1360 1370 1400 1410 1430
        1450 1450 1460 1490 1490 1500 1510 1650 1660 1670 1700 1710 1730 1740 1750
        1820 1830 1830 1860 1870 1870 1940 1950 1960 1960 1970 1990 2000 2000 2000
        2010 2010 2020 2070 2080 2090 2110 2130 2150 2160 2180 2190 2230 2250 2270
        2280 2290 2300 2310 2320 2330 2340 2350 2360 2360 2380 2380 2390 2460 2470
        2470 2480 2490 2500 2510 2530 2540 2570 2580 2590 2600 2620 2630 2670 2700
        2700 2730 2760 2760 2830 2840 2850 2880 2900 2910 2910 2920 2950 2960 2970
        2980 3010 3150 3160 3170 3200 3210 3210 3240 3250 3260 3290 3290 3300 3330
        3340 3350
K$      1080 1080 1090 1090 1100 1100 1110 1110 1240 1240 1250 1250 1260 1260 1270
        1270 2750 2760 2790 2800 3110 3120
K$()    130  2580 2590 2760 3340
K9      120  130  2570 2730 3330
L       230  240  290  350  360  370  2580 2590
L$()    140  2000 2010 2470 2470 3160 3200 3250 3300
L1      820  830  850  860  880  890
L3      1050 1960 3210 3210 3290 3290
L6      1390 1410 1560 1600 1900
M       760  770  770  780  790  820  900
N$      60   90   90   200
P       520  530  580  580  690
P3      2220 2230
P4      160  2000 2010 2040 2210 2240 2280 2290 2300 2310 2320 2330 2340 2360 2370
        2400 2530
R()     140  1330 1330 1340 1340 1350 1350 1360 1360 1410 1430 1460 1500 1660 1700
        1710 1740 1820 1860 2290 2300 2310 2320 2330 2340 2350 2360 2360 2380 2380
        2500 2510 2530
R4      1010 1490 1490 1870 1870 2090
R5      1010 1450 1450 1830 1830 2110 2150
S$      250  310  310  350  380  510  530  590  690  970  990  1150 1190 1390 1800
        2590 2630 2700 2840 2860 2880 2960 2970 2990 3010 3180
S()     110  660  670  830  830  840  850  850  860  1550 1600 1620 1620 1630 1900
S1()    110  660  670  870  880  880  890  1700 1710 1780 1790
S5      110  1540 1550 1580 1580 1600 1620 1620 1630 1700 1710 1780 1790 1890 1900
        1900 1900
S6      110  650  650  660  660  670  670  760  790  1540 1890
T       550  610  1550 1560 1570 1570 3050 3080
T$      530  570  590  620  2630 2640 2650 2660 2700 2750 3060 3070
V$      1110 1270 2770 2910 2910 3200 3250
```

Table 1. Index of all variables used in the flowcharting program.

an empty (blank) location. In this case, column L8 is chosen since L$(8) is empty—then line 3200 places an I into L$(8). This occupies that column. Then line 3210 updates a pointer L3, which indicates the leftmost column just occupied; this indicates how far left the dashes have to go later. If there is a NEXT in this line, lines 3230 through 3310 look at the variable after the word NEXT and search through L$( ) for that marker. When the routine finds the marker, it erases it from L$( ) and again updates pointer L3. This handling of FOR-NEXT statements is simple.

Returning to line 1080, the next few lines do some simple editing. A few extra characters are added to the STOP, END and GOSUB keywords; then the entire keyword is tacked onto the line number in C$.

Fig 1. FLOGEN divides each line into these columns.

In case of multiple statements on the same line, separated by a colon delimiter, lines 1140–1280 go through the same process again to pick up other keywords, tack them on to C$ and update the left, or FOR-NEXT, column markers some more. This kind of processing correctly takes care of even unusual combinations such as a FOR and its NEXT in the same line.

Lines 1320 and on take care of the transfer arrows on the right. Just as L$( ) keeps track of FOR-NEXT columns on the left, so R( ) keeps track of transfer columns on the right. Since there are ten R columns, R( ) is dimensioned as R(10), with each location corresponding to a column. If R( ) for a particular column is negative that column needs up arrows; if it is positive and greater than 1 it needs down arrows. The absolute value of that entry indicates how far down (to what line number) the arrows should go. An entry of 0 indicates that the column is empty and can be reused. An entry of 1 or slightly less indicates that the column is empty but cannot be reused since it was used in the previous line; I call this "partially released." This is important because a column reused on the next line after it is released will have two sets of arrows without a space between them.

Lines 1320–1370 scan through all of R( ), gradually releasing a column.

Lines 1390–1510 check all occupied columns to see whether they should be released in the current line. Since R( ) holds the bottom line number of the transfer associated with each column, we need only compare the contents of R( ) with the current line number. If they match, it is time to release a column. R4 and R5 are then updated to mark how far left or right arrows should go on that line, and the column is partially released when either a 1 or a .9 is placed into the appropriate R( ) location. This eventually determines whether a < or an ↑ is printed in that column.

Line 1540 checks whether this line is the beginning of a transfer down or the end of a transfer up. Since array S( ) has addresses of all transfers, lines 1540–1600 look at the next element of S( ). If the current line number is in that array, a transfer starts.

Several transfers can share the same column—if they all have the same destination and go in the same direction. In this case, lines 1650–1710 find such a column; otherwise, lines 1720–1760 find an empty column, and then lines 1780–1870 put a marker into R( ). Line 1860 puts an extra .5 into R( ) to indicate that this is the top of an upward transfer; this will then change the printing of an arrow and immediately be subtracted, in lines 2360–2380, when it's time to print the line. Lines 1890–1900 check if there are more transfers below and repeat the procedure if there are. This is necessary in ON . . . GOTO statements, which may have several destinations from one line.

Once the various column markers are set up, the rest of the program, lines 1910–2420, is straightforward, except for lines 2290–2360, which handle some exceptions at the corners of a GOTO or THEN set of arrows. This ensures that these arrows don't point the wrong way when they

change direction from vertical to horizontal or vice versa.

Finally, lines 2440–2550 close the file and check for any leftover markers indicating an unfinished FOR-NEXT loop or a downward transfer that started at some GOTO or THEN and never reached a destination.

Modifications

Assuming that your printer has about 72 columns, the program is set up for up to ten nested FOR-NEXT loops on the left, and for up to ten down or up transfers on the right. It's not likely that you'll have more than ten nested FOR-NEXT loops, but it is possible to run out of space on the right for transfers.

If your printer can print more than 80 columns across the page, you can expand the transfer columns to as much as 20. Find all the FOR K=1 TO 10 and FOR K=10 TO 1 STEP −1 statements that refer to transfer columns and change the 10 to 20; the DIM statement in line 140 is already set up for that.

This version of FLOGEN runs in a 32K system with about 22K RAM left over after the BASIC interpreter. In smaller systems, remove all REMs and reduce the size of arrays S( ), S1( ), R( ) and H$( ). Reducing the size of S( ) and S1( ) allows fewer transfers in a program before the arrays overflow. The present limit of 250 transfers is too big for most programs anyway.

FLOGEN purposely does not plot transfer arrows for GOSUB statements to cut down on the complexity of the flowchart. If you want your GOSUBs plotted, add the following statement:

```
0415 IF MID$(S$,I,5)="GOSUB" THEN B=5
     : GO TO 510
```

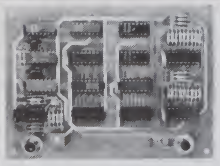
and delete lines 1100 and 1260.

This version of the program has 32 keywords, but you can easily add or remove keywords; just change the DATA statements at the end and change K9 to fit. Three asterisks should always mark the end of the keyword table.

FLOGEN is complex and takes time to run. If you want to keep tabs on what it is doing, add some additional printouts. One useful one is 0225 PRINT A$.

Although this program is written for a particular BASIC, feel free to modify it for your own system. It produces interesting and worthwhile documentation that may be useful if you have to modify a program in the future.■

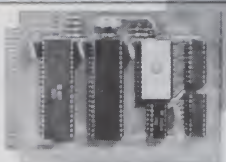
A-D & D-A CONVERTER



JBE A-D & D-A Converter can be used with any system having parallel ports • Interfaces with JBE Parallel I/O Card • D-A conversion time — 5 µs • A-D conversion time — 20 µs • Uses JBE 5V power supply • Parallel inputs & outputs include 8 data bits, strobe lines & latches • Analog inputs & outputs are medium impedance 0 to 5 volt range.

79-287 ASSM. **$79.95**
Bare Board **$39.95** Kit **$59.95**

6502 MICROCOMPUTER



This control computer has: • 1024 bytes RAM (two 2114s) • 2048 bytes EPROM (2716) • Uses one 6522 VIA (comp. doc. incl.) • Interfaces with JBE Solid State Switches & A-D & D-A Converter • Uses JBE 5V power supply • 2716 EPROM available separately (2716 can be programmed with an Apple II & JBE EPROM Programmer & Parallel Interface) • 50 pin connector included in kit & assm.

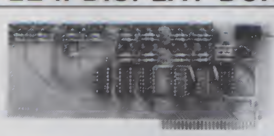
80-153 ASSM. **$110.95**
Bare Board **$49.95** Kit **$ 89.95**

SOLID STATE SWITCH

Your computer can control power to your printer, lights, stereo & any 120VAC appliances up to 720 watts (6 amps at 120VAC). Input 3 to 15VDC • 2-14MA TTL compatible • Isolation — 1500V • Non zero crossing • Comes in 1 or 4 channel version • Includes doc. for interfacing with Dimmer Control.

79-282-1 ASSM. **$13.95**
Bare Board **$6.95** Kit **$10.95**
79-282-4 ASSM. **$49.95**
Bare Board **$24.95** Kit **$39.95**

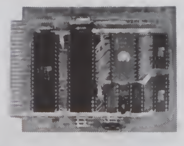
APPLE II DISPLAY BOARD



80-144 ASSM. **$49.95**
Bare Board **$25.95** Kit **$42.95**

• Has run-stop, single step switch • Has 16 address LEDs, 8 data LEDs & 1 RDY LED • All lines are buffered.

Z80 MICROCOMPUTER



JBE is announcing a single board dedicated computer designed for control functions. It features: • A Z80 Microprocessor software compatible with the Z80, 8080 & 8085 Microprocessors • Uses a Z80 PIO chip for I/O which has 2 independent 8 bit bidirectional peripheral interface ports with handshake & data transfer control • Uses one 2716 EPROM (2K) & two 2114 RAM memories (1K) • Single 5V power supply at 300MA req. • Clock frequency is 2MHz, RC controlled • Board comes with complete doc. • 50 pin connector is included • 2716 EPROM available separately.

80-280 ASSM. **$129.95**
Bare Board **$49.95** Kit **$119.95**

PRINTER INTERFACE

JBE Parallel Printer Interface interfaces your Apple II® to Centronics® compatible printers. This 3" x 4" board features: on board ROM compatible with Integer Basic, Applesoft® and Pascal® • Has one 8 bit parallel latched output port with selectable positive or negative strobe and one bit input selectable for Ready or Ready • Cable and Connectors available separately.

80-297 ASSM. **$79.95**
Kit **$69.95**
4 ft. Std. Dip Jumpers 16 pin **$ 4.25**
Champ Connector **$ 9.95**

POWER SUPPLIES

• Use wall transformers for safety • Protected against short circuit and thermal breakdown.

5 VOLT POWER SUPPLY

Rated at 5V 500MA • Operates JBE A-D & D-A Converter, Z80 & 6502 Microcomputers, 8085 & 8088 Microcomputers.

80-160 ASSM. **$20.95**
Bare Board **$8.95** Kit **$16.95**

± 12 VOLT POWER SUPPLY

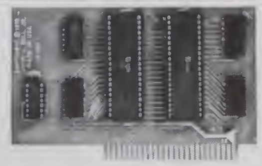
Rated at ± 12V 120MA • Can be used as a single 24V power supply • Ideally suited to OP-AMP experiments.

80-161 ASSM. **$22.95**
Bare Board **$8.95** Kit **$18.95**

ICS

6502	**$9.95**
6522	**$9.95**
Z80	**$9.95**
Z80 PIO	**$9.95**
2716 5V	**$14.95**

6522 APPLE II INTERFACE



• Interfaces printers, synthesizers, keyboards, JBE A-D & D-A Converter & Solid State Switches • Has handshaking logic, two 6522 VIAs & a 74LS74 for timing. Inputs & outputs are TTL compatible.

79-295 ASSM. **$69.95**
Bare Board **$39.95** Kit **$59.95**

2716 EPROM PROGRAMMER

JBE 2716 EPROM Programmer was designed to program 5V 2716 EPROMS • It can also read 2716s. It interfaces to the Apple II using JBE Parallel I/O Card & four ribbon cable connectors • An LED indicates when power is being applied to the EPROM • A textool zero insertion force socket is used for the EPROM • Comes with complete doc. for writing and reading in the Apple II or Apple II + • Cables available separately.

80-244 ASSM. **$49.95**
Bare Board **$24.95** Kit **$39.95**

BARE BOARDS

APPLE II EXTENDER BOARD

3½" x 2½". Price includes 50 pin Apple Connector.

80-143 **$12.95**

8085 3 CHIP SYSTEM

State-of-the-art system using an 8085, 8156 & either an 8355 or 8755 • Instruction set 100% upward compatible with 8080A.

Bare Board **$24.95**

8088 5 CHIP SYSTEM

An 8086 family microcomputer system using an 8088 CPU, 8284, 8155, 8755A & an 8185.

Bare Board **$29.95**

CRT CONTROLLER

This intelligent CRT Controller is based on an 8085A Microprocessor & an 8275 Integrated CRT Controller. It features: • 25 lines, 80 characters/line • 5x7 dot matrix • Upper case only • Two 2716s • Serial Interface RS232 & TTL • Baud rates of 110, 150, 300, 600, 1200, 2400, 4800 & 9600 • Keyboard scanning system • Req's. unencoded keyboard • Uses +5V & ± 12V power supplies.

Bare Board **$39.95**

50 pin connector **$5.95**
STD. Dip Jumpers 16 Pin, 2 ft. **$4.25**

JOHN BELL ENGINEERING ✓99

MC

ALL PRODUCTS ARE AVAILABLE FROM **JOHN BELL ENGINEERING** • P.O. BOX 338
REDWOOD CITY, CA 94064 • ADD 6% SALES TAX IN CALIFORNIA • ADD 5% SHIPPING & HANDLING
(415) 367-1137

VISA

Here's a handy little utility that keeps track of the free (unused) sectors on an Apple II floppy disk.

Apple Sector Counter Extraordinaire

By Craig Peterson

The message DISK FULL can be irritating to an Apple II owner when it flashes right in the middle of saving a long file of data or a long program. It would be nice to know how many free (unused) sectors are available. This short machine-language program, called Free Sectors, will provide you with this utility.

Inside Apple's disk operating system (DOS), there lives a friendly beast called VTOC (volume table of contents). Put simply, it contains information which tells the DOS what parts of the floppy disk already have information stored on them and what parts are available for use. Actually, the floppy disk stores this information in a special place, but the DOS reads it into a special VTOC save place (buffer) just about every time it does anything with the disk. VTOC tells how many free sectors you have.

Operation

The operation of Free Sectors (FS) consists of five main steps: save the 6502 registers, read the floppy disk's VTOC into the VTOC buffer in DOS, count the free sectors, print the free sectors message and restore the 6502 registers. The program uses some of the machine-language routines built into the Apple's monitor program, and also some that are part of the DOS. FS is assembled to run at location $0300 (hexadecimal) to keep it out of the way of any other programs you might have in memory (see Listing 1).

Also, FS is set up to run on DOS version 3.2, 3.2.1, or newly released 3.3, and for a memory size of 48K RAM. If you have less than 48K, then the addresses for PRT#, RVTC and VPTR must be adjusted downward according to your memory size. For example, if you have 32K RAM, PRT# becomes hexadecimal $6E46 (it was hexadecimal $AE46). Similarly, the addresses for RVTC and VPTR become $6FF7 and $73F2, respectively.

Before you can use the program, you must load it into memory. If you don't have an assembler program to type the program into, the simplest method is to use the Apple's monitor program. From either BASIC, type CALL −151 to get into the monitor. Then enter the hexadecimal code of the program directly into memory at location $0300 beginning with "48 98 48 8A . . .," etc.

After entering the program it's a good idea to check it by listing it in the monitor and comparing that listing with the original. If everything looks OK, BSAVE it to your disk using whatever name you desire. The starting address is $0300 and length is

Listing 1. Free Sectors machine-language program for the Apple II.

```
                    0010 :********************************
                    0020 :*                              *
                    0030 :*         FREE SECTORS         *
                    0040 :*              BY              *
                    0050 :*        CRAIG PETERSON        *
                    0060 :*         OCTOBER 1980         *
                    0070 :*                              *
                    0080 :*  A PROGRAM TO COUNT THE FREE *
                    0090 :*  (UNUSED) SECTORS AVAILABLE  *
                    0100 :*  ON AN APPLE II FLOPPY DISK  *
                    0110 :*                              *
                    0120 :********************************
                    0130 :
                    0140 CNTR .DL 0044       SECTOR COUNTER
                    0150 PRT# .DL AE46       PRNT NUM IN $44
                    0160 RVTC .DL AFF7       READ VTOC SUBR
                    0170 VPTR .DL B3F2       VTOC POINTER
                    0180 CRTN .DL FDBE       PRINT CARIJ RTN
                    0190 COUT .DL FDED       PRINT CHARACTER
                    0200 :
                    0210      .OR 0300       PROGRAM ORIGIN
                    0220 :
0300  48            0230 STRT PHA            SAVE THE 6502
0301  98            0240      TYA             REGISTERS
0302  48            0250      PHA
0303  8A            0260      TXA
0304  48            0270      PHA
0305  20F7AF        0280      JSR RVTC       READ VTOC->BUFR
0308  A900          0290      LDA 00         ZERO OUT THE
030A  8544          0300      STA *CNTR       SECTOR COUNTER
030C  8545          0310      STA *CNTR+01
030E  A28C          0320      LDX 8C         SET X INDEX
0310  A008          0330 LUP1 LDY 08         SET Y = 8 BITS
0312  BDF2B3        0340      LDA VPTR,X     GET VTOC BYTE
0315  0A            0350 CBIT ASL             AND COUNT BITS
0316  9006          0360      BCC SKIP
0318  E644          0370      INC *CNTR
031A  D002          0380      BNE SKIP
031C  E645          0390      INC *CNTR+01
031F  88            0400 SKIP DEY
031F  D0F4          0410      BNE CBIT
0321  CA            0420      DEX
```

More

Address correspondence to Craig Peterson, 1743 Centinela, #102, Santa Monica, CA 90404.

$62. Now you can quickly BLOAD it whenever you want to use it.

Using Free Sectors is simple. After loading it into memory, just insert the disk you're interested in and type CALL 768. The disk will begin spinning, and a message on the video screen will tell you the number of free sectors left on that disk. For another disk, insert the disk and type CALL 768 again. You can even use FS in Applesoft or Integer BASIC programs.

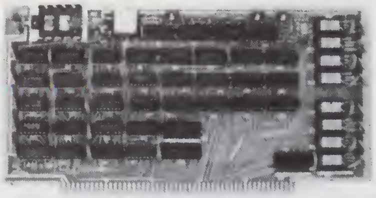
Free Sectors is a handy little utility. It probably should have been included as part of the disk operating system, but it wasn't.■

SYMBOL	TABLE
CNTR	0044
PRT#	AE46
RVTC	AFF7
VPTR	B3F2
CRTN	FD8E
COUT	FDED
STRT	0300
LUP1	0310
CBIT	0315
SKIP	031E
LUP2	0324
PRNT	0338
LUP3	0341
WRDS	0354
END	0362

Listing 1 continued.

```
0322  D0EC     0430        BNE  LUP1
0324  A545     0440  LUP2  LDA  *CNTR+01  PREPARE THE
0326  F010     0450        BEQ  PRNT       HUNDREDS PLACE
0328  38       0460        SEC             IN THE X INDEX
0329  A544     0470        LDA  *CNTR      FOR PRINTING
032B  E964     0480        SBC  64         LATER ON
032D  8544     0490        STA  *CNTR
032F  A545     0500        LDA  *CNTR+01
0331  E900     0510        SBC  00
0333  8545     0520        STA  *CNTR+01
0335  E8       0530        INX
0336  10EC     0540        BPL  LUP2
0338  208EFD   0550  PRNT  JSR  CRTN      PRINT THE FREE
033B  A002     0560        LDY  02         SECTOR COUNT
033D  8A       0570        TXA
033E  2046AE   0580        JSR  PRT#
0341  C8       0590  LUP3  INY            PRINT THE WORDS
0342  B95403   0600        LDA  WRDS,Y     'FREE SECTORS'
0345  0980     0610        ORA  80
0347  20EDFD   0620        JSR  COUT
034A  C98D     0630        CMP  8D
034C  D0F3     0640        BNE  LUP3
034E  68       0650        PLA            RESTORE 6502
034F  AA       0660        TAX             REGISTERS
0350  68       0670        PLA
0351  A8       0680        TAY
0352  68       0690        PLA
0353  60       0700        RTS
               0710  ;
0354  204652   0720  WRDS  .AS  ' FREE SECTORS'
0357  454520
035A  534543
035D  544F52
0360  53
0361  8D       0730        .HS  8D
               0740  ;
               0750  END   .EN
```

Super Compuprism Color Graphics



For the S-100 Bus. 32K of on board memory allows a 288 H. x 192V. dot matrix, for a total of 55,296 pixels. Every pixel is programable in any one of 16 colors or 16 grey levels completely independent of all other pixels in the matrix.

Compuprism Bare Board with documentation $45, kit $240, ass. and tested $280.
(16K Memory 144H. x 192V.)

Super Compuprism Bare Board with documentation $50, kit $350, ass. and tested $395.
(32K Memory 288H. x 192V.)

Add $15 to A & T price for 16 level grey scale.
Add $15 to A & T price for memory management port.

Compuprism software package, includes alpanumberics, point plot, line draw, and TRS-80* graphics simulation $20 or FREE with A & T unit.

Z-80 Users You Can Use TRS-80* Software

We offer an assembled hardware interface which we guarantee will load data from TRS-80* cassettes into any Z-80 based system. (Except sealed units.) The documentation explains how to patch the TRS-80* software to your system. In fact you can virtually change your Z-80 machine into a TRS-80* without making a single hardware change. The documentation also includes an example of patching SARGON II** into a Z-80 system.

The price is $30 or FREE with the purchase of an assembled compuprism or super compuprism unit.

A-D, D-A Board

S-100 board provides 16 channels of analog to digital input and 8 channels of digital to analog output. With on board kluge area. Total cost of board and parts less than $120. Bare board with documentation $45.

ALL COD ORDERS SHIPPED WITHIN 72 HOURS. 4MHz MOD FOR S.D. SYSTEMS.
EXPANDORAM $10.

J.E.S. GRAPHICS, P.O. Box 2752
Tulsa, OK 74101, (918) 742-7104

TRS-80* is a trademark of Tandy Corp.
SARGON II** is a trademark of Hayden Book Co.

✓180

Super Special
Apple II 16K
$1075.
reg. 1195.00

INTEGRAL DATA SYSTEMS

445G:	Paper Tiger with Graphics; 2K Buffer	$795 reg. $1095
460:	Word Processing Quality	$895 reg. 1295
460G:	IDS 460 w/Graphics	$1199 reg. 1395

Centronics 737 High Quality Dot Matrix	$895 reg. 995.00
Apple Silentype Includes interface and graphic capabilities	$535 reg. 595.00
Apple Parallel Int.	$160 reg. $180
Apple Serial Int.	$175 reg. $195
Centronics Parallel Int.	$185 reg. $225

DOUBLE VISION	$295.00
DISK II	
with controller	$580.00
without controller	$470.00
MICROMODEM	$325.00
PASCAL	$425.00
LEEDEX MONITOR	$140.00
KG-12C Green Phosphor 12" Screen w/Glare Cover 18 MHz bandwidth	$275.00

16K RAMS for APPLE II TRS-80 $59

VERBATIM DISKS 10 for $27

The Computer Stop
16919 Hawthorne Blvd.
Lawndale, CA 90260
(213) 371-4010

✓105

MON. - SAT. 10-6

MARK GORDON COMPUTERS

DIVISION OF MARK GORDON ASSOCIATES, INC.
P.O. Box 77, Charlestown, MA 02129
(617) 491-7505

✓239

COMPUTERS

Atari 800 W16K	799.00
Level-II 16K System	659.00
Model-II 64K System	3499.00
16K Model III	**879.00**

DISK DRIVES

40 Track 5¼ Inch drive	314.00
80 Track 5¼	544.00
4 Disk Drive Cable	39.00

PRINTERS

Centronics 730	599.00
Epson MX80B	Call for price
Epson MX70	Call for price
Centronics 737	849.00
Okidata Microline 83	999.00
Okidata Microline 82	729.00
Integral Data 440G	999.00
NEC 5510 w-tractor	2679.00
Okidata Microline 80	499.00
Diablo 630	2495.00

MISC HARDWARE

Expansion Int. TRS-80 (Ok)	269.00
Novation Cat modem	159.00
16K Memory Kit	41.99
Leedex Monitor	119.00
Leedex 100 G	139.00
Printer Cable for above	49.00
ISO-2 Isolator	54.00
AC LINE FILTER	24.00

STORAGE MEDIA

Verbatim box 10-5¼	25.00
Memorex box 10-5¼	22.00
Plastic Storage Box	5.00

OPERATING SYSTEMS

NEWDOS by APPARAT INC.	49.00
NEWDOS+ by APPARAT INC.	99.00
MMS FORTH DISKETTE PRIMER	79.95
NEWDOS 80	149.00

BUSINESS SOFTWARE FOR THE TRS-80* BY SBSG

Free enhancements and upgrades to registered owners for the cost of media and mailing. 30 day free telephone support. User reference on request.

Fully Interactive Accounting Package. General Ledger. Accounts Payable, Accounts Receivable and Payroll Report Generating	
Complete Package (requires 3 or 4 drives)	$475.00
Individual Modules (requires 2 or 3 drives)	$125.00
Inventory II: (requires 2 or 3 drives)	$ 99.00
Mailing List Name & Address II (requires 2 drives)	$129.00
Intelligent Terminal System ST-80 III:	$150.00
The Electric Pencil from Michael Shrayer	$150.00
File Management System	$ 49.00

FINE PRINT

TRS-80 is a Tandy Corporation trademark. Use of above operating systems may require the use of Radio Shack* TRS-DOS. Radio Shack equipment subject to the will and whim of Radio Shack.

ORDERING INFORMATION

We accept Visa and Mastercard. We will ship C.O.D. certified check or money orders only with a 40% deposit on orders over $300. Massachusetts residents add 5 percent sales tax. Shipping charges not included in prices.

To order call toll-free 1-800-343-5206
For information call 617-491-7505

The Company cannot be liable for pictorial or typographical inaccuracies.

✓Reader Service—see page 210

While not overly abundant, software for your newly assembled H-89 is available.

Building the H-89—Part 2

By Martin Moore

Software comes to the H-89 second-hand. That is, the H-89 was designed to operate with Heath's H8 software.

The H8 monitor (PAM/8) was originally written by the Wintek Corporation in 1976. In 1979 Heath updated the monitor for the H-89, keeping all the entry points and entry and exit conditions for PAM/8. In this way, software written for the H8 is directly transferable to the H-89. This is true for Extended Benton Harbor BASIC, EDIT, ASM and DBUG. All are modified versions of the Wintek H8 code. And, sadly, all are written mostly in 8080 language. Heath didn't take much advantage of the Z-80.

For $30, you get a cassette and a reference manual. On the cassette are Benton Harbor Extended BASIC, TED-8 (editor), HASL-8 (assembler) and BUG-8 (debugger). For $150, you get a disk and a reference manual. (See author's note.) The disk contains the same programs, with disk file-handling capabilities. The manuals are not for beginners. If you want tutorial information, you'll have to look elsewhere.

Both the cassette-based and disk-based programs have pretty much the same features. In the following text, I'll examine the common features of each, and discuss the particular differences between the tape and disk programs.

The Operating Systems

If you were to get the H-88 (an H-89 without disk drive), you'd require the cassette-based software. There is no operating system, per se. If you want BASIC, you position your tape to the BASIC file and use the monitor to LOAD the tape. Then you type GO.

The disk-based operating system, called HDOS, is flexible, and has been compared to CP/M (although the comparison might be stretching it a bit). In the next few paragraphs I'll discuss HDOS and its features.

Bootup—When you want to boot the H-89, you turn the machine on, insert a disk, and enter B on the terminal. The H-89 accesses the disk, then asks you to "Type SPACES to determine baud rate." This is done because HDOS will perform terminal I/O with whichever I/O port an ASCII space character comes from. In other words, if you wanted to use an auxiliary terminal, you could run the H-89 from that terminal by typing spaces at bootup.

Once you've established which terminal you're going to use, HDOS asks what you want to do: ACTION? <BOOT>. You can boot by entering a return, or you can ask the system to perform a checksum verification of all disk sectors by entering a C (for CHECK).

When you boot, HDOS tells you a number of things:

```
SYSTEM HAS 32K OF RAM
Volume 002, Mounted On SY0:
Label: PROGRAM DISK
HDOS Version 1.6
Issue #50.05.00
Date (DD-MMM-YY)?
```

As you can see, HDOS determines the amount of RAM installed in the H-89, gives you the volume number of the disk, as well as the drive (SY0:, SY1: or SY2:), and the disk label. Then the HDOS version and issue numbers are given. If the boot was from a power-up, HDOS wants to know the date. If it was a reboot, HDOS prints the date and asks you to verify it.

System Generation—You receive HDOS on a distribution disk. The first thing to be done is copy the system files onto another disk. There are three steps to copying the distribution disk. The first is initialization. The second involves running diagnostics on the H-89 and disk drive.

The diagnostics perform several important tests:

- a disk drive rotational speed test
- a general drive checkout
- a disk sector validity test
- a drive seek time test

The rotational speed test is the same one run when the H-89 is put together. It makes sure the disk rotates at the right speed. The general drive checkout is rather exhaustive, and takes over 30 minutes to run. It checks the capabilities of the drive to seek and read/write. The sector validity test checks each sector of the initialized disk and tells you how many bad sectors were discovered and which ones they are.

The drive seek time test is interesting and fairly valuable. Heath specifies that their drives will have 30 ms maximum track seek time. Heath also says that most drives will beat that time, and this test is used to find out just how fast the drive in the H-89 is. The test performs seek tests, starting at 30 ms, then works its way down in decrements of 2 ms, until it finds the

Address correspondence to Martin Moore, 2735 S.W. 229, Aloha, OR 97006.

The Heath H-89.

fastest reliable seek time. The best seek time for the drive in my unit turned out to be 10 ms. This number is important later on when the system is optimized.

Once you've run the diagnostics, you're ready to create a system disk. This is done with the SYSGEN command. All the files necessary for a minimum system are copied from the distribution disk to your new system disk.

If you want to copy BASIC, the assembler, editor or any of the other programs on the distribution disk, and you have only one drive, you'll have to use the ONECOPY program. ONECOPY is a utility that lets owners of systems with only one disk drive copy files from disk to disk.

What's Available—The distribution disk contains several files of use to the H-89 owner. Of course, BASIC, the assembler, the editor and the debugger are on the distribution disk and must be copied if you're going to use them. In addition, all the device driver programs (necessary to communicate to peripherals like printers and other terminals) are available to be copied. There's also a program called PATCH.ASM that's available, although SYSGEN doesn't copy it, and the Heath documentation doesn't say a thing about it. I assume it's used to upgrade software to newer versions.

HDOS Files

HDOS will access a file from one of three disk drives, called SY0:, SY1: and SY2:. Each file name can contain up to eight characters, as long as the characters are A through Z or 0 through 9. The file name can't begin with a number. In addition, each file name can have a three-character extension, such as .ASM or .BAS.

When files are called out in a command, they can be referenced to a specific drive (SY1:FILENAME.EXT, for example). If no drive reference is given, HDOS assumes you're talking about SY0:, the drive built into the H-89.

File manipulation might require that you specify a device in the command. Devices are those pieces of hardware that the file might be sent to or read from. For example, to copy a file to the terminal, you might enter the command TYPE FILENAME. EXT. HDOS would assume that the file was on SY0:. Or, you could enter COPY TT:=FILENAME.EXT. That would tell HDOS to copy the file to device TT:, which is the terminal. If you wanted to send a file from SY2: to the line printer, you would enter COPY LP:=SY2:FILENAME.EXT. Or, to copy a file from one drive to another, you would use COPY SY1: FILENAME.EXT=SY2:FILENAME. EXT. In this way, you can move a file almost anywhere. The recognized HDOS device names are:

SY0: Drive 0
SY1: Drive 1
SY2: Drive 2
TT: The H-89 terminal
AT: An alternate terminal
LP: The line printer
ND: The null device, used during troubleshooting, is not very useful during normal operations.

If you specify AT: or LP: in an HDOS command, you must have an appropriate program on the disk. AT: and LP: are actually device driver programs. The HDOS distribution disk contains several device driver programs that must be used to communicate with the outside world. For example, the HDOS distribution disk contains a program called LPH 14.DVD. If you want to use Heath's H-14 line printer, you must transfer the file LPH14.DVD to your disk, and rename it LP.DVD. Then, when you enter a command like COPY LP:= FILENAME.DOC, HDOS uses LP. DVD to drive the printer port.

HDOS Commands

There are 11 basic HDOS commands:

CAT—Short for Catalog, this command prints the disk directory. If you just enter CAT, the nonsystem file names are displayed on the terminal. If you enter CAT/S, all file names are displayed on the terminal. The directory listing shows the file name, extension, file size in sectors, date created or last updated and the flags. You can set the file flags to write-protect a file, suppress the file from being listed with just the CAT command or lock a file so that it can't be deleted. The CAT command can also be replaced by a DIR command. It does the same thing.

RUN—This command causes programs with the .ABS file name extension to be executed. For example, RUN EDIT causes the Heath editor to be run. You can just enter the file name, without entering RUN, and as long as the file exists with an .ABS extension, it will be executed.

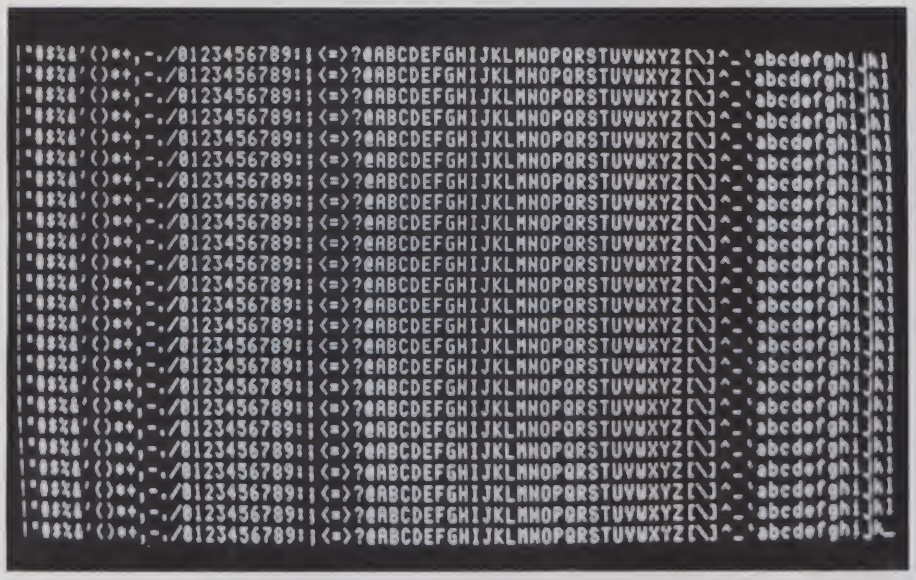
DATE—This command lets you look at the date you entered when you booted up. The DATE command also lets you change the date.

COPY—The COPY command is used to copy files to the terminal, to disk or to peripheral devices. The COPY syntax is:

COPY NEWFILE.EXT=OLDFILE.EXT

The COPY command is used to send files to peripherals by replacing the NEWFILE.EXT part of the command with a specific device driver, i.e., COPY DD:=FILENAME.EXT. DD: is my Diablo printer driver routine.

TYPE—This command types files to the terminal.



A portion of the H-89 character set. The distortion on the sides is due to photographer error, and is not apparent on the screen.

RENAME—The RENAME command does just what its name implies. It's used to rename files.

DELETE—The DELETE command also does what its name implies. DELETE can be used to delete several files at once by separating file names with commas; i.e., DELETE FILE1. EXT, FILE2.EXT, etc.

MOUNT—The MOUNT command is used to mount disks on drives.

DISMOUNT—The DISMOUNT command allows you to dismount disks in an orderly fashion.

VER—This command gives you the version of HDOS you're running.

BYE—The BYE command allows an orderly dismount of the system disk and asks for a new system disk to reboot.

STAT—The H-89 keeps track of the number of disk reads and writes you do during a session. When you enter the STAT command, HDOS tells you the number of reads and writes, as well as the number of soft (recoverable) and hard (nonrecoverable) errors. HDOS specifies that a hard error is an error that has recurred ten times during the same type of operation on the same disk.

All of these commands run under another program called PIP (Peripheral Interchange Program), the program that actually executes all the other HDOS commands. You can call up PIP simply by entering PIP. The commands used by PIP are much-abbreviated, but perform the same actions as the "higher-order" HDOS commands.

That's HDOS. Now, let's look at the other programs you get with the HDOS disk.

EDIT

Let's start with the text editor, called EDIT by HDOS and TED-8 by the cassette-based software. EDIT is a rather primitive line editor, with some limitations and some good points. By the way, there are some peculiar differences between cassette-based TED-8 and disk-based EDIT. In the following discussions of EDIT, some commands will be noted to be available in cassette software only. Heath's reason behind this is unknown to me.

EDIT is a line-oriented editor. It's always aware of the top and bottom of the text currently in the buffer. Therefore, all commands are in reference to either the top or bottom of text. For example:

```
--↑+5,$-4DELETEB
```

means "starting with the sixth line from the top (indicated by the up-arrow and +5), to, but not including, the bottom four lines (indicated by the $-4), delete all text." The "B" at the end of the command string means display the lines before they're deleted.

This is the reason I said that EDIT is a rather primitive editor. To use the type of command string shown, you have to know where you are and where you're going at all times. This type of command entry requires some care, and is difficult, but not impossible, to use.

Following are the EDIT commands. There are some file-handling commands used with the cassette-based software, but those will not be discussed here. Suffice it to say that the cassette file-handling commands are quite usable, and get the job done.

INSERT—The INSERT command allows text to be entered into the buffer space. Text can be inserted at the end of the current file, or inserted between existing lines of text.

REPLACE—This command allows an existing line or lines to be replaced.

DELETE—The DELETE command allows a single line, or a series of lines, to be deleted from the buffer.

EDIT—This is a search and replace command. The text string to be found is entered, followed by the replacement string. A single word or an entire line may be replaced. You can also specify how many occurrences of the string are to be replaced.

PRINT—This command is used to print a number of lines on the terminal. The number of lines can be qualified. Cassette software also uses this command to output text on a line printer, and to set the size of the pages printed.

TAB (cassette only)—The TAB command allows you to set tab stops in the text. Disk-based software doesn't support this command.

BLITZ—This command destroys all the text in the buffer. If you enter BLITZ, the H-89 comes back to ask if you're sure you want to do that.

SEARCH—The SEARCH command is used to search for a character string. If cassette-based software is used, SEARCH will also search through other files for the string.

NEWIN (disk only)—The NEWIN command prepares EDIT to read a new input file. The file name is given inside double quotes: NEWIN"filename.doc".

READ (disk only)—The READ command is used to read the disk file into the buffer.

NEWOUT (disk only)—The NEWOUT command prepares the disk to accept the text from the buffer. It must be followed by the file name: NEWOUT"filename.doc".

BYE—The BYE command writes the current buffer out to the disk, and then exits from EDIT.

One of the major problems with the disk-based editor is its lack of concern for the user. Let's take the following situation, for example. Suppose you have a file on disk that uses 30 sectors. There are currently 28 free sectors remaining on the disk.

If you pull the file into the editor, add one word, and then try to replace the file, EDIT will give you an error message and a terrible headache. You get the headache because EDIT requires a reserve of disk space equal to the size of the file you're trying to store. In other words, you can't replace (replace, not create) a 30-sector file if you have only 28 unused sectors. And, the worst of it is that after you attempt such a thing, it's more than likely that you'll lose any changes you made to the file because it will no longer exist in the buffer.

ASM

The Heath assembler (called ASM by HDOS, and HASL-8 by the cassette software) is a pretty good assembler. ASM has a good deal of flexibility and a lot of useful features. But—and this is the big one, folks—ASM will not assemble Z-80 code! It's incredible but true. You would think that Heath went to the Z-80 because of that microprocessor's expanded instruction set. If it were just 2 MHz operation they were after, they could have stuck with the 8080. But, there it is. An 8080 assembler. Ah, well.

The Heath Users' Group (HUG) does sell a Z-80 assembler for the H-89, however. So all is not lost.

ASM is a two-pass assembler, and will work with decimal, octal, split octal, hexadecimal or binary numbering systems. ASM recognizes five operators: +, −, *, / and − (unary negation).

Let's talk about the strength of any assembler, the pseudo-ops.

The ASM assembler supports 20 pseudo-ops, or directives. They are:

DB—Define byte. This pseudo-op defines byte contents. The bytes can be defined as integers, labels or character strings.

DW—Define word. The DW pseudo-op defines 16-bit word constants.

DS—Define space. This pseudo-op reserves a block of memory for use by a labeled symbol. For example, "BUFF DS 80" reserves 80 bytes of memory for BUFF.

IF/ELSE—The IF and ELSE directives are conditional assembly directives. The IF pseudo-op will conditionally prevent assembly of the lines following IF, so long as the argument after the IF directive is false. IF is used very much like it is in BASIC. ELSE turns assembly back on. In other words, if line 20 of your source code contained an IF directive, and the argument was false, no more code would be assembled until ASM found the ELSE statement.

ENDIF—The ENDIF pseudo-op means "don't assemble any more lines if the argument following ENDIF is false."

END—This pseudo-op marks the end of the program.

EQU—Equate. The EQU pseudo-op assigns an arbitrary value to a symbol in the label field.

SET—This pseudo-op, like EQU, assigns an arbitrary value to a symbol in the label field. Unlike EQU, however, the label defined by SET can be redefined later in the program.

ORG—Origin. The ORG statement sets the initial value of the memory location counter.

ERRxx—There are four conditional error pseudo-ops. They are:

- ERRZR—tests for a zero expression
- ERRNZ—tests for a non-zero expression
- ERRPL—tests for a positive expression
- ERRMI—tests for a negative expression

When the assembler encounters one of these error statements, it checks the expression following the statement. If the expression is true, nothing happens. If the expression is false, an error code is flagged in the assembly listing.

The following pseudo-ops control the way the assembler listing is printed out.

TITLE—The TITLE statement lets you create a new page title.

STL—Subtitle. This sets a new subtitle for the page.

EJECT—Starts a new page.

SPACE—This statement is used to set the top and bottom margins of a page.

LON—Listing on. This statement sets the listing options, which include:

- list all program lines
- list only lines containing errors
- list the lines skipped by the IF pseudo-op
- list all generated bytes

LOF—Listing off. This statement disables all the LON options.

These assembler commands are available in both ASM and HASL-8.

The disk-based ASM has a special command called XTEXT, which lets the assembly-language programmer make use of some of the HDOS functions by allowing "system calls" to HDOS software. For example, there is an HDOS routine that reads characters from the terminal. That routine (called .SCIN) can be called by your assembly-language program if you enter "SCALL .SCIN" at the point that you want to read a terminal character.

One further requirement is that you tell ASM that you're going to require the services of system calls by placing the XTEXT command at the beginning of your program. The XTEXT command is followed by the name of a file with an *.ACM (assembler common)* extension. Heath provides a file called HDOS.ACM that's used with XTEXT. The HDOS.ACM file contains a few basic system call pointers. If you want to access disk files with an assembly-language program, you have to write Heath and ask for a manual called the *HDOS System Programmer's Guide*. The manual is free of charge and contains the system calls that are used to access disk files.

DBUG

The Heath Console Debugger (DBUG for the disk-based system, and BUG-8 for the cassette-based system) is a pretty handy tool for sorting out those errors that were left in your

program following assembly. The debugger allows you to:

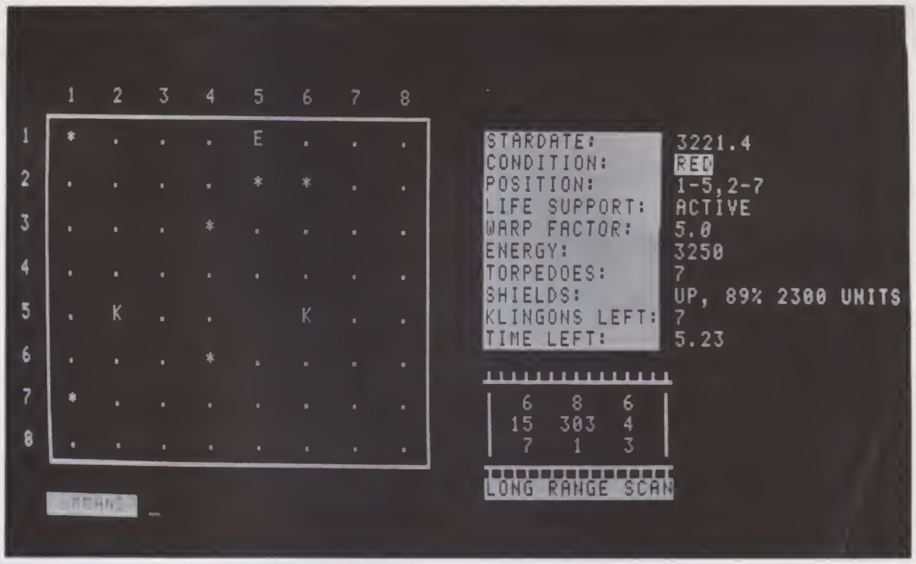
- load a program from tape or disk
- look at and change memory contents
- look at and change the 8080 equivalents of Z-80 registers
- set breakpoints in the program
- single-step program execution
- execute the program
- dump the modified program back onto tape or disk

Ignoring the program load and dump, let's look at what you can do with DBUG.

Firstly, you can look at and alter memory using either octal, decimal or ASCII notation. The notation selection is done with the first field of a memory command. Secondly, you can specify a range to be displayed or altered by entering a character in the second field of the command. For example:

:B:D1200–1205 331 201 009 131 167 309

In the command listed, :B: is the prompt character. The D tells DBUG to display the memory contents in decimal notation, byte format. And the 1200–1205 gives a memory range. The six values following the command and range are the contents of memory, listed in decimal notation. If you changed the D to an F, the memory contents would have been listed in octal notation, word format.



A familiar display, using the H-89 graphics characters and reverse video. The H-89's addressable cursor allows you to update a display like this without rewriting the entire screen.

Memory can be displayed in a variety of ways, and in a variety of ranges. For example, if the 1200–1205 had been replaced with /25, the 25 addresses following the last address of the previously specified range would have been displayed. Lots of flexibility here.

Memory is altered with an = sign.

Reader Service—see page 210

If you enter an address followed by an = sign, DBUG will display the contents of that address followed by a slash. Then you can place a new value into memory. For example:

:B: 60000 = 000/101

In this example, 60000 was entered as the address of concern. DBUG displayed the 000/ as the current contents of address 60000. The 101 following is the new value to be entered into 60000.

As I said before, DBUG is a handy tool. But, you can't look at or alter all the Z-80 registers. You can only look at the 8080-type registers. Again, what a pity that Heath didn't make full use of the Z-80. And did you notice I didn't mention hexadecimal displays? Sorry, but DBUG doesn't do hex.

Extended Benton Harbor BASIC

What can I say about Benton Harbor BASIC? It's like a lot of other medium-quality BASICs. It will handle numeric and string data, Boolean values, variables and subscripted variables, expressions, your normal arithmetic operators, relational operators, Boolean operators and so on.

This BASIC has two modes of operation: command mode and program mode. In some cases, these two modes can be mixed, as we'll see later.

Command mode statements are statements you can use without assigning them a program line number. The command mode statements are:

BUILD—This statement is used to build a program. Some other BASICs refer to this as auto number. The BUILD statement automatically supplies the next line number each time you hit return. For example, "BUILD 50,10" means "starting at line 50, automatically create new lines in increments of 10 each time a return is entered."

DELETE—This statement lets you delete a single program line, or a range of lines. For example, "DELETE 170,250" deletes lines 170 through 250.

SCRATCH—SCRATCH is a little more violent than DELETE, in that it destroys the entire program. When you enter SCRATCH, BASIC comes back and asks if you're sure you want to do that.

CONTINUE—This is an interesting statement. If your program contains a STOP statement, the H-89 will, of course, stop. But you can continue past the STOP statement by entering CONTINUE. This is great for debugging programs. Liberally sprinkle STOPs thorughout the program, and Extended Benton Harbor BASIC will let you regain control, take a look at your variables and continue on your merry way, without starting all over again.

LOCK—The LOCK statement keeps you from accidentally changing or running a program you've loaded from disk or tape. LOCK prevents execution of all the command mode statements we've discussed up to now, plus some of the file-handling statements. LOCK is implemented automatically each time a program is loaded.

```
10 CNTRL 0,100
20 FOR I = 1 TO 7
30 PRINT "X","X","X","X"
40 NEXT I
50 END
100 PRINT "WOOPS!"
110 RETURN

* RUN

X       X       X       X
X       X       X       X (<CNTRL> B TYPED)

WOOPS!
X       X       X       X
X       X       X       X
X       X       X       X
X       X       X       X (<CNTRL> B TYPED)

WOOPS!
X       X       X       X

END AT LINE 50
*
```

Listing 1. CNTRL 0 statement.

UNLOCK—What you have to do to run a locked program.

There is a second set of statements that falls between command mode statements and program statements. These statements can be used within a program, or they can be used outside the program. For example, when in the command mode, you can enter the following statements:

```
*Print "Hello There"
*Hello There
*
```

or

```
*FOR I = 1 TO 3:PRINT "Hello There!";:NEXT I
*Hello There! Hello There! Hello There!
*
```

As you can see, there's no line number. As soon as BASIC is loaded and running, you can enter a variety of commands without actually creating a line-numbered program.

The second half of the example shows another feature of Extended Benton Harbor BASIC. You can enter multiple statements on a line, as long as they're separated by a colon. In this particular mode, you can only enter as many statements as will fit on a single line. Statements requiring more than one line must be placed into a line-numbered program.

Let's look at the statements you can use in this way. Remember, all of these statements can be used in a program, or outside a program.

CLEAR—The CLEAR statement will clear the contents of a single variable, or the contents of all variables, arrays, string buffers and stacks.

CNTRL—The CNTRL statement is a multifunction command. There are five CNTRL options:

- **CNTRL 0.** This option points to a subroutine that will be executed every time <CNTRL> B is entered on the terminal. Each time <CNTRL> B is entered, the executing program is interrupted, and control is passed to a subroutine pointed to by the CNTRL 0 statement. An example of the CNTRL 0 statement is shown in Listing 1. The subroutine at line 100 could be whatever you wanted it to be.
- **CNTRL 1.** The CNTRL 1,n command sets the number of digits permitted before exponential notation is used. The value of n is from one to six digits.
- **CNTRL 2.** This command is a holdover from the H8, and is used to control the H8 front panel LED display. It serves no useful purpose in the H-89.
- **CNTRL 3.** The CNTRL 3 statement controls the size of the print zones used in a PRINT statement.
- **CNTRL 4.** This command turns the 2 ms clock within the H-89 off and on. The clock is used in clock-dependent functions such as PAUSE and PAD.

DIM—This statement is used just like it would be in a normal program —it dimensions variables. DIM is one of those statements you can use to change variable sizes after your program has stopped. You can put a STOP in the program, and once the program has ceased execution, go in and alter the variable, and then CONTINUE.

FOR/NEXT—The FOR and NEXT statements are used in exactly the same manner as shown in the first example.

GOSUB—You can enter a GOSUB statement in the command mode, and follow it with the CONTINUE statement, and program execution will begin where you told it to.

GOTO—The GOTO statement works in the same way as the GO-

SUB statement.

IF/THEN—These statements are used just as they are in the program. If a variable you enter in command mode is true, then the statement following the THEN will be executed.

LET—The LET statement also works just as it does in a program.

LIST—List one line, a range of lines or the entire program.

ON GOSUB—This statement works exactly the same as it would if it were in a program.

ON GOTO—Again, this statement works exactly the same as it would if it were in a program.

OUT—This statement sends a binary number to an output port specified in the command.

PAUSE—This is another interesting command. PAUSE will let you delay execution of the program. For example, if you just enter PAUSE, either in the program or in command mode, program execution will stop until you hit another key. If you enter PAUSE followed by an integer, program execution will be paused two times the integer (in ms).

POKE—This statement pokes a decimal number into a selected memory address.

PEEK—The PEEK statement lets you look at the contents of a selected memory location.

PORT—The PORT statement assigns control functions away from the original console. In other words, you could connect another terminal to the H-89 and direct BASIC to talk to that terminal instead of the built-in terminal.

PRINT—This statement is used just as it is in a program.

READ—The READ statement can be used in the command mode, although it is of limited value because the DATA statement has to follow on the same line.

REM—Again, this statement is of little value in the command mode, but it's available.

RESTORE—The RESTORE statement moves the data pointer back to the beginning of a DATA list.

STEP—The STEP statement is the single-step command. You can tell BASIC to execute n number of lines and then halt and wait for another STEP command. Very handy for troubleshooting your program.

FREE—This statement tells you the condition of your program and memory. FREE gives you the following information:

- bytes used by the program text
- bytes used by variables and arrays
- bytes used by FOR loops
- bytes used by GOSUBs
- bytes used by expression and function evaluation
- bytes used by STRINGs
- total number of free memory bytes remaining

All of these statements can be entered on the terminal without a line number. They're executed as soon as the return is entered.

Program Mode Functions

There are only a few remaining statements that cannot be entered except in a program. They include DEF FN, END, INPUT, LINE INPUT and STOP. A complete list of statements, as well as the predefined functions used in Extended Benton Harbor BASIC, is shown in Table 1.

There is a peculiar difference between disk BASIC and cassette BASIC. Cassette BASIC has a "command completion" feature. That is, you only have to type in the first few

letters of a command, and cassette BASIC will fill in the rest of the command. For some strange reason, disk BASIC doesn't have that feature.

Another thing that's missing from the BASIC package is a built-in renumbering routine. You can't renumber a program unless you write a renumbering program yourself . . . or buy one from somebody.

As you can see, Extended Benton Harbor BASIC is complete. You can do most anything you want. It doesn't, however, strike me as being particularly fast. The North Star Horizon BASIC has about ten times the execution speed. But, the H-89 is about on par with Radio Shack's TRS-80 Model I.

Software Support

Heath has been accused of not providing a great deal of software for its machines. And, it's true that there aren't nearly as many people developing software for the H-89 as there are for the TRS-80. But software is available.

One source of software is the Heath User's Group. The group has a software library of over 500 programs for the H8, H11 and H-89. For more information, you can write to the Heath Users' Group, Hilltop Road, Saint Joseph, MI 49085.

Then there's BUSS: The Independent Newsletter of Heath Co. Computers. The people at BUSS provide a monthly newsletter concerning the activities of Heath and the users of Heath equipment. As far as early information is concerned, the BUSS folks seem to have a pipeline directly to the Heath Company. You can get more information from BUSS, 325 Pennsylvania Ave. S.E., Washington, DC 20003.

Lifeboat Associates is offering a Heath CP/M operating system for the H-89. Their asking price is $145. The CP/M package includes a text editor, assembler, debugger and other utilities, and you get six user's manuals. Lifeboat Associates says that most programs designed to run under CP/M will work with this version, including MicroSoft BASIC, FORTRAN and COBOL.

Heath itself is expanding its software line somewhat. Heath offers MicroSoft BASIC, MicroSoft FORTRAN and the Zenith Electronic Typing word processor. Heath also offers and supports version 2.2 of CP/M on the H-89. Heath plans to offer an eight-inch floppy disk system (double-sided, double-density, IBM-compatible), too.

Author's note: New additions to the Heath line of software include 0-based CP/M and a new version of HDOS.

Conclusions

I hope I haven't sounded too disparaging about the H-89. I like it. And, I hope this has given you a better look at the H-89. This is a powerful computer, with a lot of software backing it up. If your requirements for a computer are the same as mine were, I'd recommend the H-89 highly. I'm pleased, and I think you would be too. ■

BUILD	BYE
CONTINUE	DELETE
LIST	OLD
REPLACE	RUN
SAVE	SCRATCH
CHAIN	CLEAR
CLOSE	CONTROL
DIMENSION	FOR/NEXT
FREE	FREEZE
GOSUB	RETURN
GOTO	IF/THEN
LET	LOCK
ON/GOSUB	ON/GOTO
OPEN	OUT
PAUSE	POKE
PRINT	READ/DATA
REMARK	RESTORE
STEP	UNFREEZE
UNLOCK	UNSAVE
DEF FN	END
INPUT	LINE INPUT
STOP	

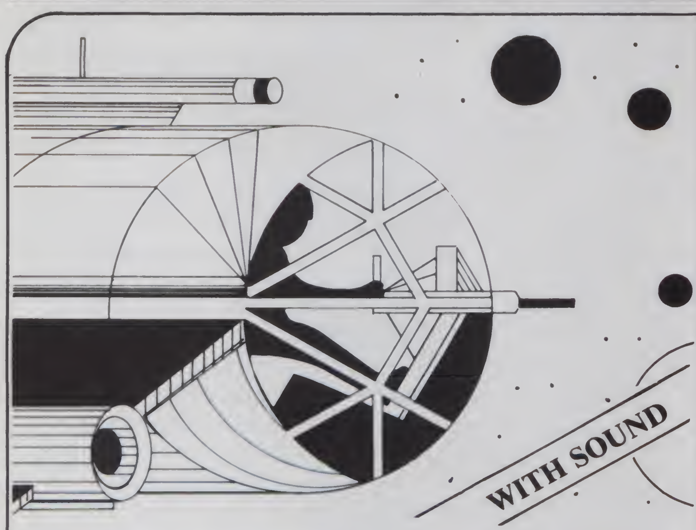
ABS	ASC
ATN	CHR$
CIN	COS
EXP	INT
LEFT	LEN
LNO	LOG
MATCH	MAX
MID$	MIN
PAD	PEEK
PIN	POS
RND	RIGHT$
SEG	SGN
SIN	SPC
SQR	STR$
TAB	TAN
VAL	

Table 1. Statements that cannot be entered except in a program and predefined functions used in B. H. BASIC.

Reader Service—see page 210

GO BOLDLY . . .

Where No TRS-80* Program Has Gone Before!



ASTEROID

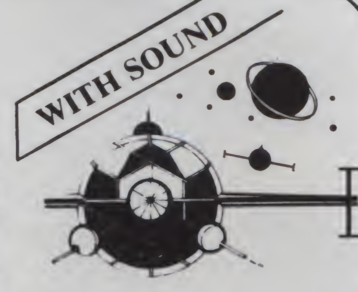
DATE: 28.02.2047
LOCATION: 270 million miles from Terra
MISSION: Maintaining Terra's Space Lanes

Briefing will follow:
1.1 Your mission is to destroy any asteroids in your sector and to prevent alien spacecraft from infiltrating the Terran Defense Network.
1.2 Your ship is armed with an anti-matter cannon. You can shoot large asteroids, but this turns them into many smaller asteroids, each capable of destroying your ship.
1.3 In addition, alien ships can make instantaneous hyperspace jumps into your area and start firing on your ship.
1.4 You'll need lightning reflexes and nerves of steel to survive Asteroid. We have no use for non-survivors!

Asteroid, a real-time, machine-language game, features variable levels of difficulty, superb high-speed graphics, sound effects and automatic score keeping. (T1) or (T2)

Order No. 0237R
$14.95 Tape.
Order No. 0247RD
$19.95 Disk.

BALL TURRET GUNNER

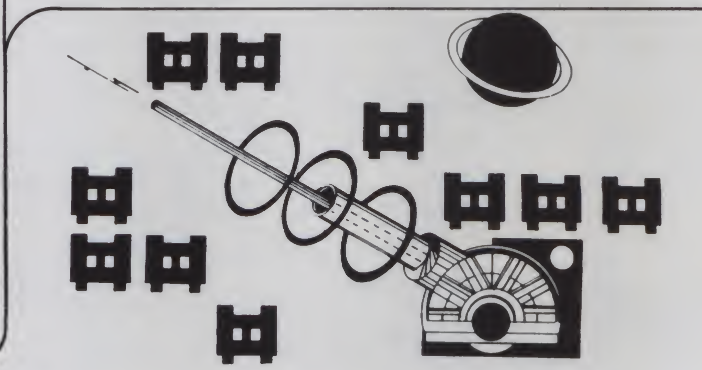


For years the Petro Resource Conglomerate has attacked our photon collection stations and strangled our deep-space trade routes. The PRC Exxonerator Class light fighters (code name: Gnat) have been their main weapon. Now you can strike back, by joining the Ball Turret Gunner Service.

Imagine yourself at the control console of an LW-1417 Stratoblazer (Type B Strategic Laser Weapon). Your Hindsight Director informs you that a Gnat fighter is coming in for an attack. You pivot your gigawatt laser turret until you can see the target on your monitor. The Range Indicator shows him coming in fast. The Targeting Computer studies his course and speed as your finger tenses over the firing key. You know you'll have only a fraction of a second in which to react. The Gnat fighter's evasive maneuvers cause him to dance in your sights. Suddenly, you see the FIRE Command and you react instinctively. Your laser beam lashes out and reduces the Gnat to an expanding ball of ionized gas. Mission accomplished!

Ball Turret Gunner, with your choice of multiple levels of difficulty, optional sound effects and superb graphics, is more than just a game. It's an adventure. Experience it! (T1)

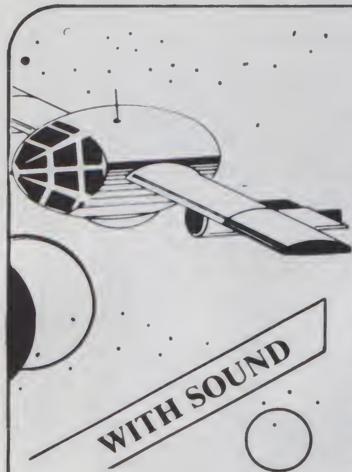
Order No. 0051R $9.95.



INVADERS

The INVADERS are coming! Earth's defenses are dead except for your Laser base. Your assignment is to destroy the approaching INVADERS before they destroy Earth. Before Earth's sensors failed, they detected 550 armed invaders in space, speeding toward us in 10 attack formations of 55 in each group. The sensors detected four different types of attack craft: Large, Medium, Small, and a short profile craft which is the most difficult to destroy. If you cannot stop these space attackers they will stop Earth..... for good. (T1)

Order No. 0240R $9.95.



COSMIC PATROL

WARNING: PLAYERS OF THIS GAME SHOULD BE PREPARED FOR A STATE OF REALISM HITHERTO UNAVAILABLE ON THE TRS-80

Skilled players soon master many difficult computer games, but COSMIC PATROL is in a world all its own. The challenge intensifies! Supporting graphics and sound (optional) make each encounter an exciting new experience. It all adds up to a Super 3-S package...skill, sight and sound.

Scenario: The Cosmic Patrol program puts you in the command chair of a small interstellar patrol craft. Your mission is to defend Terran space and prey on the Quelon supply ships which carry essential parts and lubricants for that implacably hostile robotic force. The drone freighters are fairly easy pickings for the accomplished starship pilot, but beware of the I-Fighter escorts. They're armed, fast and piloted by intelligent robots linked to battle computers. They *never* miss.

The Cosmic Patrol program is not just another search and destroy game. With its fast, real-time action, impressive sound option and superb graphics, this machine-language program is the best of its genre.

Don't keep putting quarter after quarter into arcade games or spending big bucks for video game cartridges. Get Cosmic Patrol from Instant Software—and get the best for less! (T1)

Order No. 0223R
$14.95 Tape.
Order No. 0224RD
$14.95 Disk.

TO ORDER

See Your Local Instant Software Dealer or Call Toll-Free 1-800-258-5473

Instant Software™ ✓40

PETERBOROUGH, N.H. 03458 603-924-7296

(T1) = TRS-80 Model 1, Level II, 16K RAM.

(T2) = TRS-80 Model 1, Level II 16K, Expansion Interface 16K + 1 disk drive.

*A trademark of Tandy Corporation

With this SWTP program, young students will enjoy learning about arithmetic.

Math Can Be Fun

By Richard R. Parry

Sample run.

```
 RUN
HI! MY NAME IS PAL, WHAT'S YOURS? JENNIFER
OK JENNIFER, LETS PLAY A NUMBERS GAME
WOULD YOU LIKE INSTRUCTIONS (YES OR NO) ? YES
I WILL GIVE YOU ADDITION, SUBTRACTION, MULTIPLICATION,
OR DIVISION PROBLEMS. A LITTLE BIT OF EACH IF YOU LIKE
SIMPLY
TYPE 1 FOR ADDITION
TYPE 2 FOR SUBTRACTION
TYPE 3 FOR MULTIPLICATION
TYPE 4 FOR DIVISION
TYPE 5 FOR A LITTLE OF EACH
WHEN YOU ARE DONE PLAYING, TYPE 999 WHEN I
GIVE YOU A PROBLEM AND I WILL GIVE YOU YOUR SCORE
DON'T FORGET TO DEPRESS THE 'RETURN' KEY AFTER EVERY ENTRY
WHAT IS YOUR PLEASURE (1,2,3,4 OR 5)? 5
WHAT MAXIMUM VALUE SHOULD I GIVE FOR ONE NUMBER ? 12
WHAT MAXIMUM VALUE SHOULD I GIVE YOU FOR THE OTHER NUMBER ? 24
HIT ANY KEY? P

 1 .  11 X0 =? 0
 CONGRATULATIONS
 2 .  0 X12 =? 12
 SORRY, TRY AGAIN JENNIFER
 2 .  0 X12 =? 0
 NICE GOING JENNIFER
 3 .  2 X2 =? 4
 VERY GOOD JENNIFER
 4 .  0 X3 =? 0
 CONGRATULATIONS
 5 .  3 +4 =? 7
 NICE GOING JENNIFER
 6 .  12 -9 =? 3
 CONGRATULATIONS
 7 .  6 /6 =? 1
 NICE GOING JENNIFER
 8 .  10 X5 =? 55
 I KNOW YOU CAN DO BETTER THAN THAT
 8 .  10 X5 =? 15
 SORRY, TRY AGAIN JENNIFER
 8 .  10 X5 =? 50
 NICE GOING JENNIFER
 9 .  40 /5 =? 9
 I KNOW YOU CAN DO BETTER THAN THAT
 9 .  40 /5 =? 8
 NICE GOING JENNIFER
 10 .  4 X6 =? 24
 NICE GOING JENNIFER
 11 .  8 +7 =? 999
 HOPE TO SEE YOU AGAIN JENNIFER
```

More

Teaching machines have been around for a long time. Their advantages are well known to educators. With the proper software, any microcomputer can be transformed into a teaching machine. Programs have been written to teach many subjects, even a computer language. The software described in this article is aimed at teaching young students the four basic arithmetic functions: addition, subtraction, multiplication and division.

Program Requirements

There are several features that should be incorporated into a teaching machine for it to be efficient and effective. Of utmost importance is the need to keep the student actively involved and interested. Little, if any, learning will take place if the student is ambivalent.

The interactive nature of BASIC provides the involvement, and the novelty of communicating with a computer and receiving feedback provides the necessary interest. In addition, interest is enhanced by the personal nature of the messages that are communicated to the student by the computer. The responses use the student's name, which helps create a friendly environment conducive to learning.

A good teaching machine should immediately and positively reward the student for a correct answer. Re-

Address correspondence to Richard R. Parry, 38W255 Deerpath Road, Batavia, IL 60510.

sponses such as "Nice going" or "Congratulations" go a long way in encouraging the child. If an incorrect answer is given, the machine should provide neutral feedback in order not to offend or discourage the student. Responses such as "Sorry, try again" and "I know you can do better than that" encourage the student to continue.

When an incorrect answer is given, the machine gives an appropriate response and then gives the student the opportunity to try the problem again. In this Math Teacher program, there are six possible responses: three to indicate a correct response and three for an incorrect response.

A teaching machine should also be easy to use. For this reason, the program starts by asking if the student would like instructions. If so, the necessary instructions to use the teaching machine are displayed.

It is also important that the student be able to alter the complexity of the problems by allowing the student to insert the maximum value that should be given for each of the two numbers that are used in formulating the problem.

The machine should also provide variety by allowing the student to select either addition, subtraction, multiplication or division problems. A random sampling of each of these categories is also possible. Because of this flexibility, you can use the program for several grade levels. For example, addition is typically taught *in first grade, subtraction* in second, multiplication in third and division in fourth.

Finally, the machine should indicate the progress of the student. This provides additional feedback (hopefully, it is positive feedback). When the student is finished answering questions, he types 999. The machine then displays the total number of different problems answered, the number of correct and incorrect answers and the percentage correct.

The program listing shown was written in SWTP BASIC, Version 2.0. Few, if any, instructions are unique to this version; it should, therefore, run on any BASIC machine with little or no modification. The program is not particularly complex and can be easily modified if necessary. Several REM lines are given throughout the program to aid in understanding the program.■

Sample run continued.

```
TOTAL NUMBER OF PROBLEMS 10
TOTAL NUMBER CORRECT 7
TOTAL NUMBER WRONG 3
PERCENTAGE CORRECT 70 %

READY
#
```

Program listing. Arithmetic program in SWTP BASIC.

```
0100 LINE= 80
0110  X=RND(1)
0120 PRINT "HI! MY NAME IS PAL, WHAT'S YOURS";
0130 INPUT N$
0140 PRINT "OK ";N$;", LETS PLAY A NUMBERS GAME"
0142 PRINT "WOULD YOU LIKE INSTRUCTIONS (YES OR NO) ";
0143 INPUT Y$
0144 IF LEFT$(Y$,1)="N" THEN 230
0150 PRINT "I WILL GIVE YOU ADDITION, SUBTRACTION, MULTIPLICATION,"
0160 PRINT "OR DIVISION PROBLEMS. A LITTLE BIT OF EACH IF YOU LIKE"
0170 PRINT "SIMPLY"
0180 PRINT "TYPE 1 FOR ADDITION"
0190 PRINT "TYPE 2 FOR SUBTRACTION"
0200 PRINT "TYPE 3 FOR MULTIPLICATION"
0210 PRINT "TYPE 4 FOR DIVISION"
0220 PRINT "TYPE 5 FOR A LITTLE OF EACH"
0222 PRINT "WHEN YOU ARE DONE PLAYING, TYPE 999 WHEN I"
0223 PRINT "GIVE YOU A PROBLEM AND I WILL GIVE YOU YOUR SCORE"
0224 PRINT "DON'T FORGET TO DEPRESS THE 'RETURN' KEY AFTER EVERY ENTRY"
0230 PRINT "WHAT IS YOUR PLEASURE (1,2,3,4 OR 5)";
0240 INPUT T
0250 PRINT "WHAT MAXIMUM VALUE SHOULD I GIVE FOR ONE NUMBER ";
0260 INPUT N
0270 PRINT "WHAT MAXIMUM VALUE SHOULD I GIVE YOU FOR THE OTHER NUMBER ";
0280 INPUT P
0290 PRINT "HIT ANY KEY";
0291 INPUT R$
0292 FOR I=1 TO ASC(R$)
0293  R1=RND(0)
0294 NEXT I
0300 IF T < 5 THEN 330
0310 LET G = 1
0320 LET T=INT(4*RND(0)+1)
0330 LET A = A + 1
0340 LET X = INT((N+1)*RND(0))
0350 LET Y = INT((N+1)*RND(0))
0360 ON T GO TO 380,420,500,540
0370 REM ******ADDITION******
0380 LET Z1 = X + Y
0390 PRINT A;".  ";X;"+";Y;"=";
0401 GOTO 610
0410 REM ******SUBTRACTION******
0420 IF X> Y THEN 460
0430 LET X1 = Y
0440 LET Y = X
0450 LET X=X1
0460 LET Z1 = X - Y
0470 PRINT A;".  ";X;"-";Y;"=";
0480 GOTO 610
0490 REM ******MULTIPLICATION******
0500 LET Z1 = X * Y
0510 PRINT A;".  ";X;"X";Y;"=";
0520 GOTO 610
0530 REM ******DIVISION******
0540 IF X > 0 THEN 570
0550 LET X = Y
0560 LET Y = 0
0570 LET L = X * Y
0580 LET Z1 = Y
0590 PRINT A;".  ";L;"/";X;"=";
0600 REM ******INPUT THE ANSWER PORTION OF THE PROGRAM******
0610 INPUT Z
0620 IF Z = 999 THEN 790
0630 IF Z = Z1 THEN 700
```

More

Program continued.

```
0640  R3=RND(0)
0642 IF R3 > .333 THEN 668
0650 PRINT "SORRY, TRY AGAIN ";N$
0660 GOTO 680
0668 IF R3 > .667 THEN 674
0670 PRINT "I KNOW YOU CAN DO BETTER THAN THAT"
0672 GOTO 680
0674 PRINT "CONCENTRATE ";N$
0680 LET F = 1
0690 ON T GO TO 390,470,510,590
0700  R2=RND(0)
0701 IF R2 > .333 THEN 728
0710 PRINT "CONGRATULATIONS"
0720 GOTO 740
0728 IF R2 > .667 THEN 734
0730 PRINT "VERY GOOD ";N$
0732 GOTO 740
0734 PRINT "NICE GOING ";N$
0740 IF F = 1 THEN 760
0750 LET B = B + 1
0760 LET F = 0
0770 IF G = 1 THEN 320
0780 GOTO 330
0790 PRINT "HOPE TO SEE YOU AGAIN ";N$
0800 LET A = A - 1
0810 LET C = A - B
0820 LET D = 100 * B/A
0830 PRINT
0840 PRINT
0850 PRINT "TOTAL NUMBER OF PROBLEMS ";A
0860 PRINT "TOTAL NUMBER CORRECT ";B
0870 PRINT "TOTAL NUMBER WRONG ";C
0880 PRINT "PERCENTAGE CORRECT ";D;"%"
0890 END
```

Data base applications are endless with this system for the SWTP.

The Ultimate Information Juggler

By Dr. Gordon Wolfe

What do wine cellars, magazine publishers, university research directors, physicians and stockbrokers all have in common? The answer is simple: they all need to store and rapidly access information.

This fact was brought home to me a while ago when I was called into the office of the associate director of research at the university where I teach. The graduate school (which houses the university research administration) was thinking about buying a small computer, and wanted my advice on what sort of equipment to look into.

I was no stranger to the office. My job requires me to spend 50 percent of my time in original research, much of which involves the scientific applications of microcomputers and microprocessors. In addition, I run a small consulting business helping people select and set up microcomputers.

I gave the director my usual advice, and told him to be careful that software for *his* particular application was available and working on the machine he chose. It was then that my curiosity got the better of me, and I asked what applications the graduate school had in mind for their possible computer. I expected the usual accounting applications, because of all the research contracts and grants that go through the office.

I was surprised to find that accounting was only a small part of what they wanted to do. Indeed, the university accounting and auditing

Dr. Gordon Wolfe is an assistant professor of physics at the University of Mississippi, University, MS 38677.

Sample 1.

```
*
OPTION LIST

0  EXIT TO BASIC
1  PRINT CONTENTS OF FILE
2  ENTER NEW FILE
3  MODIFY ENTRIES IN A FILE
4  SEARCH ON TITLES
5  DELETE A FILE
6  ADD ENTRY TO FILE
7  DELETE ENTRY FROM FILE

OPTION NUMBER ? 2
FILE NAME (NO EXT) ? WINLST
DRIVE NUMBER (0 OR 1) ? 0
TOTAL NUMBER OF ENTRIES ? 5
COLUMNS OF DATA EACH ENTRY ? 7
COLUMN 1
DATA TYPE:ALPHA, NUMBER, OR DATE ? A
COLUMN TITLE ? VARIETY
COLUMN 2
DATA TYPE:ALPHA, NUMBER, OR DATE ? A
COLUMN TITLE ? VINTNER
COLUMN 3
DATA TYPE:ALPHA, NUMBER, OR DATE ? N
COLUMN TITLE ? YEAR
COLUMN 4
DATA TYPE:ALPHA, NUMBER, OR DATE ? D
COLUMN TITLE ? BOUGHT
COLUMN 5
DATA TYPE:ALPHA, NUMBER, OR DATE ? N
COLUMN TITLE ? COST/B
COLUMN 6
DATA TYPE:ALPHA, NUMBER, OR DATE ? N
COLUMN TITLE ? #BOT
COLUMN 7
DATA TYPE:ALPHA, NUMBER, OR DATE ? A
COLUMN TITLE ? TYPE
ENTRY NO. 1
ENTER VARIETY
 ? CABERNET
ENTER VINTNER
 ? MARTIN
ENTER YEAR
 ? 1973
ENTER BOUGHT
DATE (MO,DAY,YR) ? 3,4,74
ENTER COST/B
 ? 4.50
ENTER #BOT
```

More

Sample 1 continued.

```
 ? 12
ENTER TYPE
 ? BURGUNDY
ENTRY NO. 2
ENTER VARIETY
 ? JOHANN.R
ENTER VINTNER
 ? MONDAVI
ENTER YEAR
 ? 1976
ENTER BOUGHT
DATE (MO,DAY,YR) ? 6,11,78
ENTER COST/B
 ? 6.10
ENTER #BOT
 ? 2
ENTER TYPE
 ? WI DELETED
WHIG DELETED
WI DELETED
WHITE
ENTRY NO. 3
ENTER VARIETY
 ? RHINE
ENTER VINTNER
 ? GALLO
ENTER YEAR
 ? 79
ENTER BOUGHT
DATE (MO,DAY,YR) ? 4,25,80
ENTER COST/B
 ? 1.54
ENTER #BOT
 ? 200
ENTER TYPE
 ? WHITE
ENTRY NO. 4
ENTER VARIETY
 ? PORT
ENTER VINTNER
 ? MASSON
ENTER YEAR
 ? 1965
ENTER BOUGHT
DATE (MO,DAY,YR) ? 7,11,75
ENTER COST/B
 ? 6.22
ENTER #BOT
 ? 1
ENTER TYPE
 ? DESSERT
ENTRY NO. 5
ENTER VARIETY
 ? NECTAR
ENTER VINTNER
 ? ALMADEN '
ENTER YEAR
 ? 78 DELETED
1978
ENTER BOUGHT
DATE (MO,DAY,YR) ? 2,15,79
ENTER COST/B
 ? 3,11 DELETED
3.11
ENTER #BOT
 ? 4
ENTER TYPE
 ? ROSE

*
OPTION LIST

0   EXIT TO BASIC
1   PRINT CONTENTS OF FILE
2   ENTER NEW FILE
3   MODIFY ENTRIES IN A FILE
```

(More

department would really take care of most of that, as they are doing now. What the graduate school wanted was a machine for the following applications:

- Word processing—writing proposals and project reports to government and private funding agencies.
- Information storage and retrieval—storing information on what applications for grants have been made, to whom, for how much, by whom, from what department, and selectively retrieving them by category. (For instance, determining how many proposals for amounts over $30,000 were made by the physics department in 1979.)
- Keeping a file on the areas of expertise of faculty members, and searching them as needed for funding of research projects. (The EPA is requesting proposals on effects of pesticides on freshwater crustaceans. Who do we have that knows anything about pesticides and/or crustaceans?)
- Keeping the graduate school's intramural and extramural mailing lists.

The last three of the above tasks are all essentially the same type of task, but with different specific applications. That is, they all store information, and then selectively retrieve it to give information on the status of specific operations.

Indeed, in large systems, this is one of the more common uses of a computer: the storage and manipulation of large amounts of data in a utilitarian and task-determined manner. Somehow, this manipulation, which is called data base management, has been more or less overlooked on the microcomputer.

Unlimited Opportunities

The possibilities of a data base management system are limited only by the imagination of the user. A few possible applications are:

- Keeping a catalog of a wine cellar.
- Keeping up with a stock portfolio.
- A directory of personnel expertise.
- A doctor's list of prescribed drugs, allergies and diagnoses for a particular patient.
- A mailing list for newsletters or Christmas cards.
- Indexing articles in back issues of your magazines or newsletters.

Some data base management programs have been written for micros. But for the most part they are oriented toward specific applications, or limit the types of data used in some way. Examples include the many mailing-list programs (usually cassette or memory-resident files), the wine list program of Rodney Jollife ("Computerized Wine Cellar," *Byte*, February 1979, p. 128) or the club management program of Thomas Doyle ("Do-All-Plus," *Microcomputing*, April 1980, p. 164). Peter Reese has written a more general data base program, which is oriented toward string searches and sorts ("Complete Data Base Management System," *Interface Age*, August 1978, p. 108), but this was still not general enough for the application I had in mind.

A complete data base management program should do the following:

1. Keep files on disk so that file size is not limited by available memory, or speed limited by the cassette recorder.
2. Allow input, printing, deletion and modification of files. (Add or delete entries or modify data within an entry.)
3. Keep data for alphanumeric strings, numerical quantities or dates (which are, after all, a set of three numbers).
4. Allow a search on a file for all entries which meet one *or more* criteria, and either print the results of the search, or make the results available for further manipulations or numerical calculations.
5. Keep titles for the categories in the file, and keep track of what types (i.e., alpha, numeric or date) of data are present in the file, and where they are.
6. Be usable in as many applications as possible, and be self-explanatory in its use.

Program Features

With these capabilities in mind, I wrote the program DATBAS (Listing 1) in Computerware Disk File BASIC version 5.5 for the Smoke Signal Broadcasting BFD-68 disk system on the SWTP 6800. It requires 5K of program space above the 11K disk file BASIC, so at least 16K of memory is required. The BASIC uses sequential disk files, so to edit files, a new file is created, the old file, with modifications, is read into it, and the old file is deleted. File size is limited only by the size of the disk.

All information is kept on a file on the disk. When asked for the name of the file, you can input any string of letters or numbers. The program will truncate the string to a maximum of six characters, and add the extension

.DAT, so that all data files will have the extension .DAT. A file contains N entries, or individual pieces of information.

Within each entry, there are N1 categories (up to 10). Each category can be alphanumeric, numeric or a date (which will be stored as a numeric quantity in the format [day + 100*month + 10000*year]. It is best to use only the last two numbers of the year.) The type of category and the title of each category, along with the numbers N and N1, are stored on disk as a file header, and are followed by the data for each entry in sequence. Once entered, the number and arrangement of categories may not be changed by DATBAS.

The program DATBAS is written as a structured program, permitting easy modifications to your system. An executive routine oversees the particular procedure being used. These procedures are written as separate routines and are called from the executive. A series of subroutines (all included in line numbers from 1000 to 2980) handles processes used by more than one process routine, such as read a data file header, or write an entry to a file, etc. All routines and subroutines have line numbers, which are even hundreds, and all are identified by plentiful remark statements.

Sample 1 continued.

```
4  SEARCH ON TITLES
5  DELETE A FILE
6  ADD ENTRY TO FILE
7  DELETE ENTRY FROM FILE
OPTION NUMBER ? 1
FILE NAME (NO EXT) ? WN DELETED
WINLST
OUTPUT ON SCREEN, PRINTER,
OR DISK ? S

        VARIETY   VINTNER  YEAR     BOUGHT    COST/B    #BOT
 TYPE

1       CABERNET MARTIN   1973     3/4/74    4.5       12
 BURGUNDY
2       JOHANN.R MONDAVI  1976     6/11/78   6.1       2
 WHITE
3       RHINE    GALLO    79       4/25/80   1.54      200
 WHITE
4       PORT     MASSON   1965     7/11/75   6.22      1
 DESSERT
5       NECTAR   ALMADEN  1978     2/15/79   3.11      4
 ROSE
PRESS RETURN TO CONTINUE
 ?
```

Upon execution of DATBAS, the executive prints the menu of services and asks for the process desired in terms of an option number.

Options

Sample 1 illustrates the option list printout and the selection of option 2, which creates a new file and sets up the category information. The sample run is an example of the use of DATBAS as a management program for a private wine cellar.

After selection of the option and entry of the file name and drive number, the program will ask for the number of entries. (In this case, the number of different types of wine purchases to be entered at this time.) Then the program will ask for the

```
*
OPTION LIST

0  EXIT TO BASIC
1  PRINT CONTENTS OF FILE
2  ENTER NEW FILE
3  MODIFY ENTRIES IN A FILE
4  SEARCH ON TITLES
5  DELETE A FILE
6  ADD ENTRY TO FILE
7  DELETE ENTRY FROM FILE

OPTION NUMBER ? 1
FILE NAME (NO EXT) ? MAILST
OUTPUT ON SCREEN, PRINTER,
OR DISK ? S
          NAME      ADDR       CITY/ST  ZIP

1         G.WOLFE  PHYSICS DEPT
                               UNIVERSITY MS
                                        38677
2         J.CARTER 1600 PENNSYLVANIA
                               WASHINGTON DC
                                        20044
3         W.GREEN  PINE ST    PETERBOROUGH NH
                                        3458
PRESS RETURN TO CONTINUE
 ?
```

Sample 2.

```
*
OPTION LIST

0  EXIT TO BASIC
1  PRINT CONTENTS OF FILE
2  ENTER NEW FILE
3  MODIFY ENTRIES IN A FILE
4  SEARCH ON TITLES
5  DELETE A FILE
6  ADD ENTRY TO FILE
7  DELETE ENTRY FROM FILE

OPTION NUMBER ? 6
FILE NAME (NO EXT) ? PATNT
HOW MANY NEW ENTRIES ? 1
ENTRY NO. 4
ENTER NAME
 ? L.BORGIA
ENTER FIRST VISIT
DATE (MO,DAY,YR) ? 1,2,34
ENTER ACCOUNT
 ? 12.50
ENTER ALLERGIES
 ? ARSENIC
ENTER PRESCRIPTION
 ? ETHYL ALCOHOL
ENTER VISIT DATE
DATE (MO,DAY,YR) ? 4,3,80
```

Sample 3.

```
OPTION LIST
0  EXIT TO BASIC
1  PRINT CONTENTS OF FILE
2  ENTER NEW FILE
3  MODIFY ENTRIES IN A FILE
4  SEARCH ON TITLES
5  DELETE A FILE
6  ADD ENTRY TO  FILE
7  DELETE ENTRY FROM FILE

OPTION NUMBER ? 1
FILE NAME (N O EXT) ? SOCKS
FILE 1:SOCKS.DAT NOT FOUND
FILE NAME (NO EXT) ? STOCKS
OUTPUT ON SCREEN, PRINTER,
OR DISK ? S
         STOCK     NO.SHARESDATE PURCHASED
                                      COST      VALUE
1        ITT       100      2/21/54   23        34
2        IBM       200      5/11/69   112       223
3        KRO       300      1/4/80    0.28      0.01
4        XXX        10       9/1/76    22        33
PRESS RETURN TO CONTINUE
 ?
*
OPTION LIST
0  EXIT TO BASIC
1  PRINT CONTENTS OF FILE
2  ENTER NEW FILE
3  MODIFY ENTRIES IN A FILE
4  SEARCH ON TITLES
5  DELETE A FILE
6  ADD ENTRY TO FILE
7  DELETE ENTRY FROM FILE

OPTION NUMBER ? 3
FILE NAME (NO EXT) ? STOCKS
MODIFY HOW MANY ENTRIES ? 3
ITEM NUMBERS TO MODIFY
 ? 3
 ? 4
 ? 1
ENTRY NO. 1
ENTER STOCK
 ? ITT
ENTER NO.SHARES
 ? 100
ENTER DATE PURCHASED
DATE (MO,DAY,YR) ? 2,21,54
ENTER COST
 ? 23
ENTER VALUE
 ? 41
ENTRY NO. 3
ENTER STOCK
 ? KRO
ENTER NO.SHARES
 ? 200
ENTER DATE PURCHASED
DATE (MO,DAY,YR) ? 1,4,80
ENTER COST
 ? .28
ENTER VALUE
 ? 0
ENTRY NO. 4
ENTER STOCK
 ? XXX
ENTER NO.SHARES
 ? 10
ENTER DATE PURCHASED
DATE (MO,DAY,YR) ? 9,1,76
ENTER COST
 ? 22
ENTER VALUE
 ? 445
```

Sample 4.

number of categories to be included in each entry in the data base.

In the example, the wine will be categorized by variety, vintner, vintage year, date of purchase, cost per bottle, number of bottles in that purchase and wine type (red, white, rose, dessert, aperitif or sparkling). There are seven categories, so seven is input in response to the prompt. For each category, the user is asked for the type of data (alpha, numeric or date) and the title of the category. Using this information, the disk file header is prepared. Then, for each entry, the information is requested by the title of the category, until all N entries have data for them.

Option 1 is also shown in the example, giving a nice neat printout of the file in table format.

Sample 2 is an example of the use of the table print option to print a mailing list. Additional categories here might include subscription expiration dates, or officer titles if this was a newsletter for an organization or club. The search option could then be used to print only nonexpired subscriptions on the printer, or only expired subscriptions so that reminder notices may be sent. The same may be said for printing a list of subscriptions about to expire in, say, the next 60 days.

Sample 3 shows how a physician might use option 6 to add a new entry to his file of patient visit records. The prompts for the entry are, again, just the titles of the categories.

Sample 4 shows the use of option 1

Sample 5.

```
*
OPTION LIST

0  EXIT TO BASIC
1  PRINT CONTENTS OF FILE
2  ENTER NEW FILE
3  MODIFY ENTRIES IN A FILE
4  SEARCH ON TITLES
5  DELETE A FILE
6  ADD ENTRY TO FILE
7  DELETE ENTRY FROM FILE

OPTION NUMBER ? 1
FILE NAME (NO EXT) ? RESRCH
OUTPUT ON SCREEN, PRINTER,
OR DISK ? S
        P.I.      AMOUNT    APPLIED FOR
                                      STATUS    AGENCY

1       A.EINSTEIN
                  300000    1/1/53    FUNDED    AEC
2       M.SNERD  1.5       2/30/60   DENIED    CIA
3       A.CAPONE 25000     9/12/29   DENIED    FBI
PRESS RETURN TO CONTINUE
 ?
```

More

Sample 5 continued.

```
*
OPTION LIST

0   EXIT TO BASIC
1   PRINT CONTENTS OF FILE
2   ENTER NEW FILE
3   MODIFY ENTRIES IN A FILE
4   SEARCH ON TITLES
5   DELETE A FILE
6   ADD ENTRY TO FILE
7   DELETE ENTRY FROM FILE

OPTION NUMBER ? 4
FILE NAME (NO EXT) ? RESRCH
SEARCH ON HOW MANY CATEGORIES ? 2
OUTPUT ON SCREEN, PRINTER,
OR DISK ? S
*
1   P.I.
2   AMOUNT
3   APPLIED FOR
4   STATUS
5   AGENCY

INPUT NUMBERS OF SEARCH CATEGORIES
 ? 3
 ? 4,
ITEM 3 APPLIED FOR
LATEST
DATE (MO,DAY,YR) ? 12,31,79
EARLIEST
DATE (MO,DAY,YR) ? 1,1,60
ITEM 4 STATUS
SEARCH STRING ? DENIED
```

to print the stock portfolio for an executive. The value of the stocks changes on an hourly basis, so the executive might run option 3 once a week or so to update the value of the stocks. Three of the entries—items 1, 3 and 4 in the list—are to have the category "present value" changed during the week of the example. Note that modification (or deletion, for that matter) is based on entry number for simplicity. Entry numbers can be input in any order; the program will sort them into ascending order. To modify an entry, the entire entry is reentered with modified data.

If a stock is doing well, a user might make an additional purchase of that stock and use option 6 to make the new entry. The search option could then be used to look at the total per-

```
0900 REM SUBROUTINE TO READ DATA FILE
0910 REM HEADER CREATED BY DATBAS.BAS
0920 REM FILE NAME IN F$ WITH EXTENSION

0930 REM .DAT
0940 REM T$ HAS TITLES OF CATEGORIES
0950 REM C HAS INFO WHETHER COLUMN J
0960 REM 1=ALPHANUMERIC
0970 REM 2=NUMERIC
0980 REM 3=DATE (DAY+100*MO+10000*YEAR)
1000 DIM T$(10),A$(10),A(10),C(10)
1010 OPEN #1,F$
1020 IF STATUS#1=0 THEN 1050
1030 PRINT "FILE ";F$;" NOT FOUND"
1040 END
1050 READ #1,X
1060 IF STATUS#1=6 THEN 1090
1070 LET N1=N:N=X
1080 GOTO 1050
1090 RESTORE #1
1100 FOR I=1 TO N
1110 FOR J=1 TO N1
1120 READ #1,X
1130 NEXT J
1140 NEXT I
1150 FOR J=1 TO N1
1160 READ #1,T$(J),C(J)
1170 NEXT J
1180 RESTORE #1
1185 LET I=1
1190 RETURN
1200 REM SUBROUTINE TO READ ENTRY I
1201 REM FROM FILE CREATED BY DATBAS.BAS

1202 REM SUBROUTINE 1000 MUST BE
1203 REM EXECUTED FIRST AND
1204 REM THIS ROUTINE CALLED FROM
1205 REM A FOR-NEXT LOOP INDEXED I
1220 FOR J=1 TO N1
1230 ON C(J) GOTO 1240,1260,1260
1240 READ #1,A$(J)
1250 GOTO 1270
1260 READ #1,A(J)
1270 NEXT J
1280 IF I=N THEN CLOSE #1
1290 RETURN
```

Listing 2. Subroutine to read the new disk file.

formance of all holdings of the stock, and to make a record of long-term capital gains at tax time.

Option 4 is the key to the entire program. Up to this point, we have merely discussed the storage and retrieval of data; now it is time to look at selective retrieval. An example of the use of option 4 is shown in Sample 5.

In this sample run, a college is keeping tabs on the proposals written for government grants. The file is printed out to show, in this example, that a total of three proposals have been written. The name of the investigator, the amount applied for, the date of application, whether the request was funded, denied, or still pending, and the funding agency are all stored in the file.

The director of research might want to know how many funding requests were denied during the 20-year period of the 1960s and 1970s. Instead of having a secretary spend days digging through the back files, he merely executes option 4 on the file RESRCH.

He wants to know two things: how many were denied and how many of these denials fell in those two decades? This requires a search on two categories: transmittal date and status. He punches in that the search is to be over two categories and specifies categories 3 and 4, which are the categories specified above. Since item 3 is a date, he must also input the range of the search—what starting and ending dates are to be looked for. S search string ''denied'' must be input for the funding status.

At this point, the program will read each entry from the file and determine if the transmittal date is within the range specified, and, if so, also determine if the ''status'' string contains the substring ''denied,'' and will print all entries which meet both these criteria. Entry no. 2 is the only one so doing, and is the only one printed.

The director could just as easily have asked for all proposals still pending from the CIA, or for all proposals submitted in 1979 from a particular department (if department had been one of the categories).

The end of a search is rarely the end of the information. Usually, some other procedure needs to be done. The research director might want to know the total dollar amount of all funds granted in 1979. Or a stockbroker might want to know the net capital gain per share of a particular stock if he sold now. A statistical analysis on the data or a sort on names might be required. All of these situations could be programmed into DATBAS, but each person's needs will be different. It would be impossible to write a general-purpose analysis program which would satisfy even a moderate percentage of data base users.

With that in mind, a provision has been made in DATBAS for the user to write his own codes for the analysis of searched data. The output of the search can be directed to disk. That is, all entries which meet the search criteria will be placed in a disk file, the name of which the user may choose, along with header information, so that results of the search can be accessed by another analysis program. The only difficulty is that the output file does not have the same format, and cannot be used as an input file in DATBAS. This was done to save processing time, as it would require two searches—one to find the total number of items which satisfy the search and one to list those items on disk.

Instead, the new disk file is created in a single pass and has a slightly different format. Title and category type is still stored. To read the new disk file, a set of subroutines is provided

Listing 1. DATBAS program for the SWTP.

```
*
0010 REM DATA BASE MANAGEMENT
0015 REM COPYRIGHT 1980
0020 REM BY GORDON W. WOLFE
0030 REM PHYSICS DEPT., UNIVERSITY
0040 REM OF MISSISSIPPI
0050 REM ALL RIGHTS RESERVED
0055 REM THIS PROGRAM MAY NOT BE COPIED
0060 REM BY ANY PROCESS, INCLUDING BUT NOT
0065 REM LIMITED TO XEROGRAPHIC OR
0070 REM ELECTRONIC METHODS WITHOUT
0075 REM WRITTEN PERMISSION FROM THE
0080 REM AUTHOR.  READERS OF KILOBAUD ARE
0085 REM GIVEN PERMISSION TO COPY FOR
0090 REM THEIR PERSONAL OR BUSINESS USE,
0095 REM BUT NOT FOR FURTHER SALE OR
0096 REM DISTRIBUTION
0099 REM ------------------------------
0100 DATA .DAT,TEMP.DAT
0110 READ E$,D$
0115 STRING= 20
0120 DIM T$(10),C(10),A(10),B(10),B$(10)S$(10),S(10,2)
0140 GOSUB 2300
0200 REM PRINT MENU
0210 PRINT CHR$(16);CHR$(11);CHR$(11);:HOME
0220 PRINT "OPTION LIST":PRINT
0230 PRINT 0;TAB(4);"EXIT TO BASIC"
0240 PRINT 1;TAB(4);"PRINT CONTENTS OF FILE"
0250 PRINT 2;TAB(4);"ENTER NEW FILE"
0260 PRINT 3;TAB(4);"MODIFY ENTRIES IN A FILE"
0270 PRINT 4;TAB(4);"SEARCH ON TITLES"
0280 PRINT 5;TAB(4);"DELETE A FILE"
0285 PRINT 6;TAB(4);"ADD ENTRY TO FILE"
0290 PRINT 7;TAB(4);"DELETE ENTRY FROM FILE"
0300 PRINT
0310 INPUT "OPTION NUMBER",O
0320 LET O=INT(O)
0330 IF O<=0 THEN END
0340 IF O>7 THEN END
0350 ON O GOTO 500,700,4000,4500,400,3000,3500
0400 REM DELETE FILE
0410 GOSUB 1000
0412 LET F=1
0415 GOSUB 1100
0417 IF N4=1 THEN  410
0420 FDEL F$
0430 PRINT "FILE ";F$;" DELETED"
0440 GOTO 200
0500 REM PRINT A FILE
0510 GOSUB 1000
0515 LET F=1
0520 GOSUB 1100
```

More

in Listing 2. To use these subroutines, first the name of the search data file with extension .DAT must be placed into the string variable F$. To initialize the file and read the header information, execute a GOSUB 1000 from your analysis routine.

Upon return from this subroutine, the array T$(J) will contain the title of the columns, with column number J. C(J) will indicate the data type of column J. C=1 if alphanumeric data is present, C=2 if the column is numeric, and C=3 if the data in the column is a date. The variable N will contain the number of entries in the table, and N1 will have the total number of columns or categories.

To read an entry from the file, execute a GOSUB 1200 from the analysis routine. This will read the next entry from the file. If no entries have been read previously, the first entry will be read. If the entry read is the last entry, the file will be closed. Upon reading the entry, data will be placed either in the string array A$(J) if the data for column J is alphanumeric, or into the numeric array A(J) if the data for column J is a numeric or date quantity. Notice that alpha and numeric data is intermixed. A$(1) can have data, but A$(2) cannot have data if column 2 is a numeric quantity—the data will be placed in A(2) instead. The array C(J) tells whether the data for column J is in A$(J) or in A(J). Be very careful with this. Also, the number of the next entry to be read from the file is kept in variable I, which should not be changed by the calling program or a disk error could result, or data might be lost.

Modifications

Modification of the program for individual needs may range from simple to quite difficult, depending on the hardware and disk file BASIC being used. Rather complete use of the provisions of Computerware's sequential disk file BASIC is made in DATBAS. The disk file is treated as a large DATA statement, so that READ and RESTORE work much the same as with a DATA statement. OPEN, CLOSE and WRITE are fairly standard disk commands. Not so standard are the FREN and FDEL commands to rename and delete files, or the STATUS function to determine if the file is open, closed, ended or nonexistent.

To use more than ten categories of data, some minor changes are required. First and foremost is the di-

Listing 1 continued.

```
0525 IF N4=1 THEN 510
0530 GOSUB 1600
0540 IF N2<>8 THEN 570
0550 PRINT "DISK COPY NOT NECESSARY"
0560 GOSUB 1600
0565 GOTO 540
0570 GOSUB 2000
0572 FOR J=1 TO N1
0574 PRINT #N2,TAB(J*9);T$(J);
0576 NEXT J
0578 PRINT #N2:PRINT#N2
0580 FOR I=1 TO N
0590 GOSUB 1400
0600 FOR J=1 TO N1
0610 GOSUB 1700
0620 NEXT J
0630 LET N3=N1
0640 GOSUB 1800
0650 NEXT I
0670 GOSUB 1900
0675 CLOSE #F
0680 GOTO 200
0700 REM ENTER NEW FILE
0710 GOSUB 1000
0720 OPEN #1,F$
0730 INPUT "TOTAL NUMBER OF ENTRIES",N
0740 INPUT "COLUMNS OF DATA EACH ENTRY",N1
0750 FOR J=1 TO N1
0760 PRINT "COLUMN ";J
0770 INPUT "DATA TYPE:ALPHA, NUMBER, OR DATE",Y$
0780 LET Y$=LEFT$(Y$,1)
0790 LET N4=0
0800 IF Y$="A" THEN N4=1
0810 IF Y$="N" THEN N4=2
0820 IF Y$="D" THEN N4=3
0830 IF N4=0 THEN 770
0850 LET C(J)=N4
0860 INPUT "COLUMN TITLE",T$(J)
0870 NEXT J
0880 LET F=1:N4=N:N5=N1
0890 GOSUB 2100
0900 FOR I=1 TO N
0910 GOSUB 2900
0960 GOSUB 1500
0970 NEXT I
0980 CLOSE #F
0990 GOTO 200
1000 REM INPUT FILE NAME
1010 INPUT "FILE NAME (NO EXT)",F$
1020 LET F$=LEFT$(F$,6)+E$
1025 IF O=2 THEN 1030
1027 IF O=5 THEN 1030
1029 GOTO 1060
1030 INPUT "DRIVE NUMBER (0 OR 1)",D
1040 IF D=1 THEN F$="1:"+F$
1050 IF D<>0 THEN 1030
1060 RETURN
1100 REM SEARCH FOR FILE
1105 LET N4=0
1110 OPEN #F,F$
1120 IF STATUS #F=0 THEN RETURN
1130 IF D=0 THEN F$="1:"+F$
1140 IF D=1 THEN F$=MID$(F$,3,LEN(F$))
1150 OPEN #F,F$
1160 IF STATUS #F=0 THEN RETURN
1170 PRINT "FILE ";F$;" NOT FOUND"
1180 LET N4=1
1190 RETURN
1200 REM CONVERT DATE TO NUMBER  D1
1210 INPUT "DATE (MO,DAY,YR)",D2,D1,D3
1220 LET D1=100*D2+D1+10000*D3
1230 RETURN
1300 REM CONVERT NUMBER TO DATE
1310 LET D3=INT(D1/10000)
1320 LET D2=INT((D1-D3*10000)/100)
1330 LET D1=INT(D1-10000*D3-100*D2)
1340 RETURN
```

More

OSI

SOFTWARE FOR OSI

OSI

A JOURNAL FOR OSI USERS!!

The Aardvark Journal is a bimonthly tutorial for OSI users. It features programs customized for OSI and has run articles like these:

1) Using String Variables.

2) High Speed Basic On An OSI.

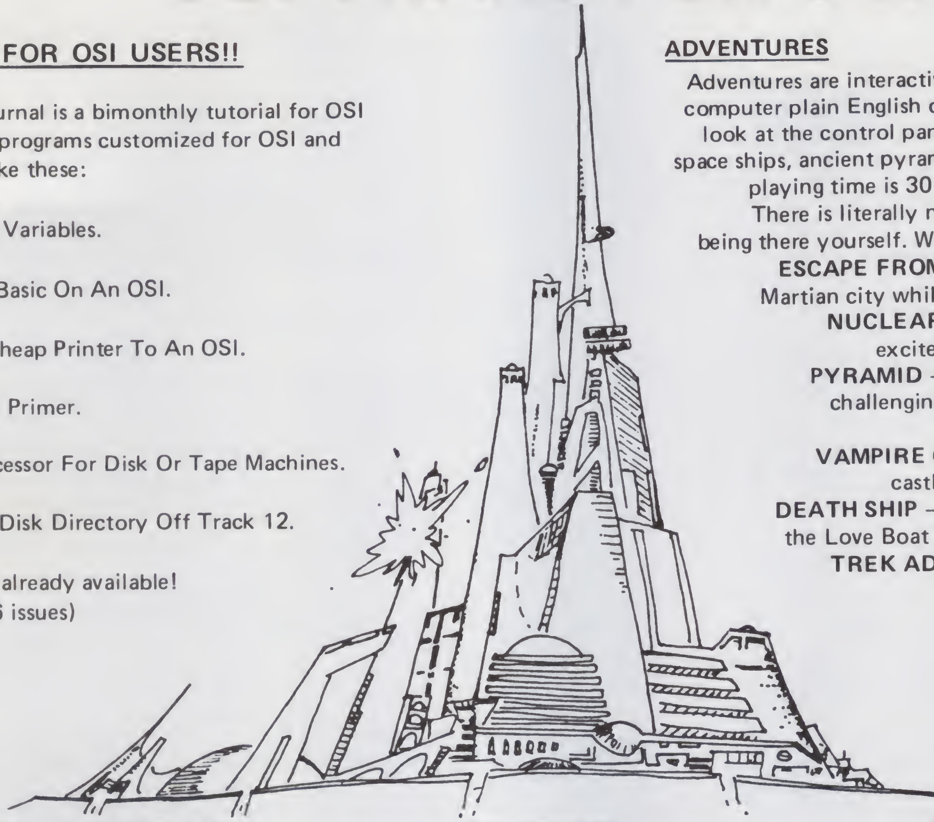
3) Hooking a Cheap Printer To An OSI.

4) An OSI Disk Primer.

5) A Word Processor For Disk Or Tape Machines.

6) Moving The Disk Directory Off Track 12.

Four back issues already available!
$9.00 per year (6 issues)



ADVENTURES

Adventures are interactive fantasies where you give the computer plain English commands (i.e. take the sword, look at the control panel.) as you explore alien cities, space ships, ancient pyramids and sunken subs. Average playing time is 30 to 40 hours in several sessions. There is literally nothing else like them — except being there yourself. We have six adventures available.

ESCAPE FROM MARS — Explore an ancient Martian city while you prepare for your escape.

NUCLEAR SUBMARINE — Fast moving excitement at the bottom of the sea.

PYRAMID — Our most advanced and most challenging adventure. Takes place in our own special ancient pyramid.

VAMPIRE CASTLE — A day in old Drac's castle. But it's getting dark outside.

DEATH SHIP — It's a cruise ship — but it ain't the Love Boat and survival is far from certain.

TREK ADVENTURE — Takes place on a familiar starship. Almost as good as being there.

$14.95 each

NEW SUPPORT ROMS FOR BASIC IN ROM MACHINES

C1S — for the C1P only, this ROM adds full screen edit functions (insert, delete, change characters in a basic line.), Software selectable scroll windows, two instant screen clears (scroll window only and full screen.), software choice of OSI or standard keyboard format, Bell support, 600 Baud cassette support, and a few other features. It plugs in in place of the OSI ROM. NOTE: this ROM also supports video conversions for 24, 32, 48, or 64 characters per line. All that and it sells for a measly $39.95.

***C1E/C2E** for C1/C2/C4/C8* Basic in ROM machines.
This ROM adds full screen editing, software selectable scroll windows, keyboard correction (software selectable), and contains an extended machine code monitor. It has breakpoint utilities, machine code load and save, block memory move and hex dump utilities. A must for the machine code programmer replaces OSI support ROM. Specify system $59.95

DISK UTILITIES

SUPER COPY — Single Disk Copier
This copy program makes multiple copies, copies track zero, and copies all the tracks that your memory can hold at one time — up to 12 tracks at a pass. It's almost as fast as dual disk copying. — $15.95

MAXIPROSS (WORD PROCESSOR) — 65D polled keyboard only - has global and line edit, right and left margin justification, imbedded margin commands, choice of single, double or triple spacing, file access capabilities and all the features of a major word processor — and it's only $39.95.

P.C. BOARDS

MEMORY BOARDS!! — for the C1P. — and they contain parallel ports!
Aardvarks new memory board supports 8K of 2114's and has provision for a PIA to give a parallel ports! It sells as a bare board for $29.95. When assembled, the board plugs into the expansion connector on the 600 board. Available now!

PROM BURNER FOR THE C1P — Burns single supply 2716's. Bare board — $24.95.

MOTHER BOARD — Expand your expansion connector from one to five connectors or use it to adapt our C1P boards to your C4/8P. - $14.95.

ARCADE AND VIDEO GAMES

ALIEN INVADERS with machine code moves — for fast action. This is our best invaders yet. The disk version is so fast that we had to add selectable speeds to make it playable.
Tape - $10.95 — Disk - $12.95

TIME TREK (8K) — real time Startrek action. See your torpedoes move across the screen! Real graphics — no more scrolling displays. $9.95

STARFIGHTER — a real time space war where you face cruisers, battleships and fighters using a variety of weapons. Your screen contains working instrumentation and a real time display of the alien ships. $6.95 in black and white - $7.95 in color and sound.

MINOS — A game with amazing 3D graphics. You see a maze from the top, the screen blanks, and then you are in the maze at ground level, finding your way through on foot. Realistic enough to cause claustrophobia. — $12.95

SCREEN EDITORS

These programs all allow the editing of basic lines. All assume that you are using the standard OSI video display and polled keyboard.

C1P CURSOR CONTROL — A program that uses no RAM normally available to the system. (We hid it in unused space on page 2). It provides real backspace, insert, delete and replace functions and an optional instant screen clear.
$11.95

C2/4 CURSOR. This one uses 366 BYTES of RAM to provide a full screen editor. Edit and change lines on any part of the screen. (Basic in ROM systems only.)

FOR DISK SYSTEMS — (65D, polled keyboard and standard video only.)

SUPERDISK. Contains a basic text editor with functions similar to the above programs and also contains a renumberer, variable table maker, search and new BEXEC* programs. The BEXEC* provides a directory, create, delete, and change utilities on one track and is worth having by itself. — $24.95 on 5" disk - $26.95 on 8".

AARDVARK IS NOW AN OSI DEALER!

Now you can buy from people who can support your machine.

—THIS MONTH'S SPECIALS—

Superboard II	$279
C1P Model II	429
C4P	749

. . . and *we'll include* a free Text Editor Tape with each machine!

Video Modification Plans and P.C. Boards for C1P as low as $4.95

This is only a partial listing of what we have to offer. We now offer over 100 programs, data sheets, ROMS, and boards for OSI systems. Our $1.00 catalog lists it all and contains free program listings and programming hints to boot.

VISA

master charge

OSI

Aardvark Technical Services • 1690 Bolton • Walled Lake, MI 48088
(313) 699-3110 or (313) 624-6316

✓91

OSI

✓Reader Service—see page 210

mension statement at line 120. There are also the 10s in lines 3552, 3554, 4090, 4080, 4570 and 4580.

All of the PRINT statements, especially the one in line 1830, may have to be modified depending on your printer and BASIC. I had considerable difficulty with print formatting, as the sample runs will attest.

Lastly, a few of the subroutines in DATBAS might be of use to you in reading search files and manipulating the data therein. Specifically, I have in mind the subroutines at lines 1300 (converts a numerical quantity to a date by changing the number stored in D1 to three numbers—D1, D2 and D3, corresponding to day, month and year, respectively) and 2600 (which sorts an array B(1) into ascending numerical order).

The purpose of a computer is to make our lives and jobs easier. One of the things they're best at is storing and sorting information. When I showed this program and the results to the associate director, his response was, "If I can get a computer to do all this for me, I won't have to hire a fifth secretary." As for me, I'm going to go catalog all the music for my banjo.■

Listing 1 continued.

```
1400 REM READ AN ENTRY FROM FILE #F (NO.I)
1410 FOR J=1 TO N1
1420 ON C(J) GOTO 1430,1450,1450
1430 READ #F,A$(J)
1440 GOTO 1460
1450 READ #F,A(J)
1460 NEXT J
1470 RETURN
1500 REM WRITE AN ENTRY TO FILE #F
1510 FOR J=1 TO N1
1520 ON C(J) GOTO 1530,1550,1550
1530 WRITE #F,A$(J)
1540 GOTO 1560
1550 WRITE #F,A(J)
1560 NEXT J
1570 RETURN
1600 PRINT "OUTPUT ON SCREEN, PRINTER,"
1610 INPUT "OR DISK",Y$
1620 LET Y$=LEFT$(Y$,1)
1625 LET N2=-1
1630 IF Y$="S" THEN N2=1
1640 IF Y$="P" THEN N2=7
1650 IF Y$="D" THEN N2=8
1660 IF N2<0 THEN 1600
1670 RETURN
1700 REM CONVERT DATA TO STRING  FOR COL J
1710 ON C(J) GOTO 1720,1740,1760
1720 LET B$(J)=A$(J)
1730 GOTO 1780
1740 LET B$(J)=STR$(A(J))
1750 GOTO 1780
1760 LET D1=A(J):GOSUB1300
1770 LET B$(J)=STR$(D2)+"/"+STR$(D1)+"/"+STR$(D3)
1780 RETURN
1800 REM PRINT ENTRY I;N3 ITEMS INB$;DEVICE N2
1810 PRINT #N2,I;
1820 FOR J=1 TO N3
1830 PRINT #N2,TAB(J*9);B$(J);
1840 NEXT J
1850 PRINT #N2
1860 RETURN
1900 REM KEEP DISPLAY UNTIL DONE
1910 PRINT "PRESS RETURN TO CONTINUE"
1920 INPUT Y$
1930 RETURN
2000 REM READ HEADER FROM FILE
2005 READ #F,N,N1
2010 FOR J=1 TO N1
2020 READ #F,T$(J)
2030 NEXT J
2040 FOR J=1 TO N1
2050 READ #F,C(J)
2060 NEXT J
2070 RETURN
2100 REM WRITE HEADER TO FILE F
2110 WRITE #F,N4,N5
2120 FOR J=1 TO N5
2130 WRITE #F,T$(J)
2140 NEXT J
2150 FOR J=1 TO N5
2160 WRITE #F,C(J)
2170 NEXT J
2180 RETURN
2200 REM RENAME TEMP.BAS TO F$
2220 IF LEFT$(F$,2)<>"1:" THEN 2270
2225 OPEN #1,D$,#2,F$
2230 LET F=1:GOSUB 2000
2233 LET N4=N1:N5=N2
2235 LET F=2:GOSUB 2100
2240 FOR I=1 TO N
2242 LET F=1
2244 GOSUB 1400
2246 LET F=2
2248 GOSUB 1500
2250 NEXT I
2255 CLOSE #1,#2
2257 FDEL D$
```

More

Listing 1 continued.

```
2260 RETURN
2270 FREN D$,F$
2280 RETURN
2300 FOR I=1 TO 10
2310 LET B(I)=0
2320 NEXT I
2330 RETURN
2500 REM DELETE F$ IF F$ ON SAME DRIVE ASD$
2510 REM D$ ALWAYS ON DRIVE 0
2520 IF LEFT$(F$,2)="1:" THEN 2550
2530 FDEL F$
2550 RETURN
2600 REM SORT B(I) INTO ASCENDING ORDER
2610 FOR K=1 TO N6
2620 LET G=1.E99
2630 FOR I1=K TO N6
2640 IF B(I1)>G THEN 2680
2650 LET G=B(I1)
2660 LET B(I1)=B(K)
2670 LET B(K)=G
2680 NEXT I1
2690 NEXT K
2695 RETURN
2900 PRINT "ENTRY NO. ";I
2910 FOR J=1 TO N1
2915 PRINT "ENTER ";T$(J)
2920 ON C(J) GOTO 2925,2935,2945
2925 INPUT A$(J)
2930 GOTO 2955
2935 INPUT A(J)
2940 GOTO 2955
2945 GOSUB 1200
2950 LET A(J)=D1
2955 NEXT J
2960 RETURN
3000 REM ADD ENTRIES TO FILE
3010 GOSUB 1000
3020 LET F=1
3030 GOSUB 1100
3040 IF N4=1 THEN 3010
3050 INPUT "HOW MANY NEW ENTRIES",N6
3060 GOSUB 2000
3070 LET N4=N+N6:N5=N1
3080 OPEN #2,D$
3090 LET F=2
3100 GOSUB 2100
3110 FOR I=1 TO N
3120 LET F=1
3130 GOSUB 1400
3140 LET F=2
3150 GOSUB 1500
3160 NEXT I
3170 FOR I=N+1 TO N4
3180 GOSUB 2900
3190 GOSUB 1500
3200 NEXT I
3210 CLOSE #1,#2
3215 GOSUB 2500
3220 GOSUB 2200
3240 GOTO 200
3500 REM DELETE AN ENTRY FROM A FILE
3505 GOSUB 2300
3510 GOSUB 1000
3520 LET F=1
3530 GOSUB 1100
3540 IF N4=1 THEN 3510
3550 INPUT "DELETE HOW MANY ENTRIES",N6
3552 IF N6<=10 THEN 3560
3554 PRINT "MAXIMUM 10"
3556 GOTO 3550
3560 GOSUB 2000
3570 LET N4=N-N6:N5=N1
3580 OPEN #2,D$
3590 LET F=2
3600 GOSUB 2100
3610 PRINT "ITEM NUMBERS TO DELETE"
3620 FOR I=1 TO N6
```

More

Listing 1 continued.

```
3630 INPUT B(I)
3640 NEXT I
3642 GOSUB 2600
3645 LET I1=1
3650 FOR I=1 TO N
3660 LET F=1
3670 GOSUB 1400
3680 IF B(I1)=I THEN 3720
3690 LET F=2
3700 GOSUB 1500
3710 GOTO 3730
3720 LET I1=I1+1
3730 NEXT I
3740 CLOSE #1,#2
3745 GOSUB 2500
3750 GOSUB 2200
3770 GOTO 200
4000 REM MODIFY ENTRIES IN A FILE
4010 GOSUB 1000
4020 LET F=1
4030 GOSUB 1100
4040 IF N4=1 THEN 4010
4050 GOSUB 2300
4060 OPEN #2,D$
4070 INPUT "MODIFY HOW MANY ENTRIES",N6
4080 IF N6<=10 THEN 4100
4090 PRINT "MAXIMUM 10":GOTO 4070
4100 GOSUB 2000
4110 LET F=2:N4=N:N5=N1
4120 GOSUB 2100
4130 PRINT "ITEM NUMBERS TO MODIFY"
4140 FOR I=1 TO N6
4150 INPUT B(I)
4160 NEXT I
4162 GOSUB 2600
4165 LET I1=1
4170 FOR I=1 TO N
4180 LET F=1
4190 GOSUB 1400
4200 IF I<>B(I1) THEN 4220
4205 LET I1=I1+1
4210 GOSUB 2900
4220 LET F=2
4230 GOSUB 1500
4240 NEXT I
4250 CLOSE #1,#2
4255 GOSUB 2500
4260 GOSUB 2200
4270 GOTO 200
4500 REM SEARCH ON MULTIPLE CATEGORIES
4510 GOSUB 1000
4520 LET F=1
4530 GOSUB 1100
4540 IF N4=1 THEN 4510
4550 GOSUB 2300
4560 INPUT "SEARCH ON HOW MANY CATEGORIES",N6
4570 IF N6<=10 THEN 4600
4580 PRINT "MAXIMUM 10"
4590 GOTO 4560
4600 GOSUB 1600
4610 IF N2<>8 THEN 4640
4620 OPEN #2,D$
4630 IF STATUS#2=0 THEN SCRATCH #2
4640 LET F=1
4650 GOSUB 2000
4660 HOME
4670 FOR J=1 TO N1
4680 PRINT J;TAB(4);T$(J)
4690 NEXT J
4700 PRINT
4710 PRINT "INPUT NUMBERS OF SEARCH CATEGORIES"
4720 FOR J=1 TO N6
4730 INPUT B(J)
4740 NEXT J
4745 GOSUB 2600
4750 FOR J=1 TO N6
4760 LET N4=B(J)
```

More

DIABLO 1620 $1895

- Letter quality daisywheel printer
- Upper and lower case
- Forms tractor, print wheel, ribbon included
- ASCII serial interface with RS-232 cable
- Bidirectional printing
- Completely refurbished by national terminal distributor with 90 day warranty

PRINTERS		VIDEO MONITORS	
NEC 5510	$2495	9" Sanyo B&W	$169
NEC 5530	2495	12" Sanyo B&W	289
Diablo 630	2425	12" Sanyo w/green screen	299
Centronics 730	675	13" Sanyo Color	495
Centronics 737	799	9" Panasonic B&W	169
		9" RCA B&W	169

DISKETTES

Verbatim			3M		
5" SS, DD	10/2.40	50/2.35	5" SS, SD	10/2.80	50/2.70
5" SS, DD	10/2.40	50/2.35	5" SS, SD	10/2.80	50/2.70
5" DS, DD	10/3.95	50/3.90	8" SS,SD	10/2.90	50/2.80
8" SS, SD	10/3.05	50/3.00	8" DS, SD	10/4.50	50/4.40
8" SS, DD	10/3.35	50/3.30	8" DS, DD	10/4.50	50/4.40
8" DS, DD	10/4.15	50/4.10	Disk Flip 'N File Case 5" 22.95, 8" 27.95		
Diskette Storage Pages 10/4.25			Disk Library Case 5" 2.05, 8" 2.80		

3M HEAD CLEANING KIT

- Eliminate downtime
- Eliminate service calls
- Increase life of read-write heads
- Cleans in 30 seconds
- Removes dust, dirt, magnetic oxides

5¼" or 8" **$19.50**

PAPER & LABELS

Greenbar 14⅞x11	
one part 3000 sheets	39.95
two part 1300 sheets	45.95
White 9½x11	
one part 3500 sheets	29.95
two part 1400 sheets	33.95
Labels 3½x15/16	
one across box of 5000	17.30
two across box of 10,000	34.60

PRINT WHEELS			RIBBONS		
Mfr.	ea.	6+	Mfr.	ea.	6+
Diablo	8.25	7.50	Diablo	6.50	72.00
Qume	8.25	7.50	Qume	4.00	43.50
NEC	17.00	15.50	NEC	7.00	71.50

Call for quantity prices—Dealer inquiries invited

DSI

DAVIS SYSTEMS INC.
2184 Meadowcliff Drive N.E., Atlanta, GA. 30345
(404) 634-2300
63

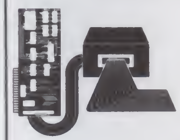
SCOTT ADAMS, President of Adventure International and author of the "Adventure" Series.

Instant Software™
PETERBOROUGH, NEW HAMPSHIRE 03458 603-9247296

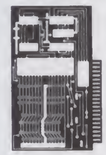
40

Measure, Control, Document...

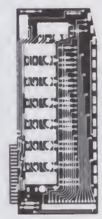
Computer Systems and Peripherals designed to coordinate with the Apple Computers that let you measure, monitor, regulate, control and document with laboratory precision, at affordable cost.



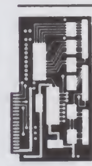
AI13 Analog Input System (A/D,12bits)
Measure voltages with 0.024% precision
Monitor 16 separate inputs under program control
Choose from 8 voltage ranges under program control; 100mV to 10V
Make a reading in 20 microseconds



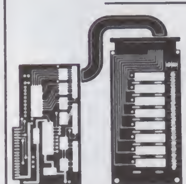
AI02 Analog Input System (A/D,8bits)
Measure voltages with 0.4% precision
Monitor 16 separate inputs under program control
Match most sensors with a standard 5 volt range
Signal Conditioning System SC14 available



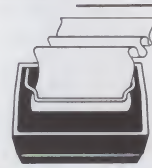
A003 Analog Output System (D/A,8bits)
Control voltage levels from a simple program
Standard 10V range, other ranges available
Up to 8 channels, each with its own memory



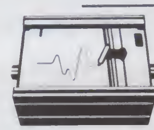
DI09 Digital Interface
Monitor or control 32 circuits in any combination
Measure time intervals or count pulses
Plug into BCD, parallel, or switch closure instrument outputs



UI16 Isolated Power Interface
Control 110VAC circuits from a program
Operate DC relays or solenoids
Receive logic pulses in noisy environments
Avoid interference and grounding problems in large systems
Start with one circuit — expand to dozens



PR12 Intelligent Printer System
Clear, readable printing on standard paper
Intermix graphics and text of three sizes
A true **printing system** including intelligent interface, and a complete manual.



PL12 Intelligent Plotter System
Produce detailed pen drawings from a program
8 x 10 Plotting surface with .005" resolution
Automatically plot from Basic arrays and strings
A true **plotting system** including intelligent interface and complete manual.

Interactive Structures has been producing laboratory systems and interfaces for Apple Computer since 1977. Our technology is in use in thousands of applications — from our own Spacelab instrumentation system to automobile engine research and woodstove analysis.

Contact us for the details on affordable laboratory computing.

iS **Interactive Structures, Inc.** 73
P.O. Box 404
Bala Cynwyd, PA 19004
215-667-1713

Listing 1 continued.

```
4770 PRINT "ITEM ";N4;T$(N4)
4780 ON C(N4) GOTO 4790,4810,4840
4790 INPUT "SEARCH STRING",S$(N4)
4800 GOTO 4900
4810 INPUT "MAX VALUE",S(N4,2)
4820 INPUT "MIN VALUE",S(N4,1)
4830 GOTO 4900
4840 PRINT "LATEST"
4850 GOSUB 1200
4860 LET S(N4,2)=D1
4870 PRINT "EARLIEST"
4880 GOSUB 1200
4890 LET S(N4,1)=D1
4900 NEXT J
4910 REM BEGIN SEARCH OF FILE
4915 LET N5=0
4920 FOR I=1 TO N
4930 LET F=1
4940 GOSUB 1400
4950 FOR I1=1 TO N6
4960 LET N4=B(I1)
4970 ON C(N4) GOTO 4980,5093,5093
4980 REM MATCH STRINGS
4990 LET L=LEN(S$(N4))
5000 LET L1=LEN(A$(N4))
5010 IF L>L1 THEN 5220
5020 LET K=1
5030 IF S$(N4)<>MID$(A$(N4),K,K+L-1) THEN 5060
5040 REM MATCH FOR THIS I1
5050 GOTO 5090
5060 IF K>=L1-L+1 THEN 5070
5063 LET K=K+1
5066 GOTO 5030
5070 REM NO MATCH
5080 GOTO 5220
5090 NEXT I1
5092 GOTO 5100
5093 REM MATCH NUMBER
5094 GOSUB 5400
5095 IF M3=0 THEN 5220
5096 GOTO 5090
5100 REM FULL MATCH ALL ITEMS
5101 REM PRINT RESULTS
5105 LET N5=N5+1
5110 IF N2=8 THEN 5190
5120 LET N3=N1
5130 FOR J=1 TO N1
5140 GOSUB 1700
5150 NEXT J
5170 GOSUB 1800
5180 GOTO 5220
5190 REM PRINT TO DISK
5200 LET F=2
5210 GOSUB 1500
5220 NEXT I
5225 PRINT I
5230 PRINT "END OF FILE ";F$
5240 CLOSE #1
5250 IF N2<>8 THEN 5295
5251 FOR J=1 TO N1
5252 WRITE #2,T$(J),C(J)
5253 NEXT J
5254 WRITE #2,N1,N5
5260 CLOSE #2
5270 PRINT "NAME FOR DESTINATION FILE"
5275 LET O=2
5280 GOSUB 1000
5290 GOSUB 2220
5295 GOSUB 1900
5300 GOTO 200
5400 LET M3=0
5405 REM IF M3=0 THEN NO MATCH
5410 IF A(N4)>S(N4,2) THEN 5440
5420 IF A(N4)<S(N4,1) THEN 5440
5430 LET M3=1
5440 RETURN
```

Find out how bidirectionality can make this simple sorting method more efficient.

A Better Bubble Sort

By W. A. Harrison and S. H. Sachs

Many methods to sort record lists are available. The bubble sort—also known as sorting by exchanging—is one of the most common.

When you sort a list of items, you must have one particular field you are sorting by. This field is known as the *sorting key*. For example, assume you are sorting a list of students by age; age would be the sorting key.

When sorting, you try to arrange your list so the values of your keys run from smallest to largest. In the process of sorting this list of items, you must at some time examine or otherwise process every key in the list. This is described as a *pass*. Each time you process the entire list, you perform a pass.

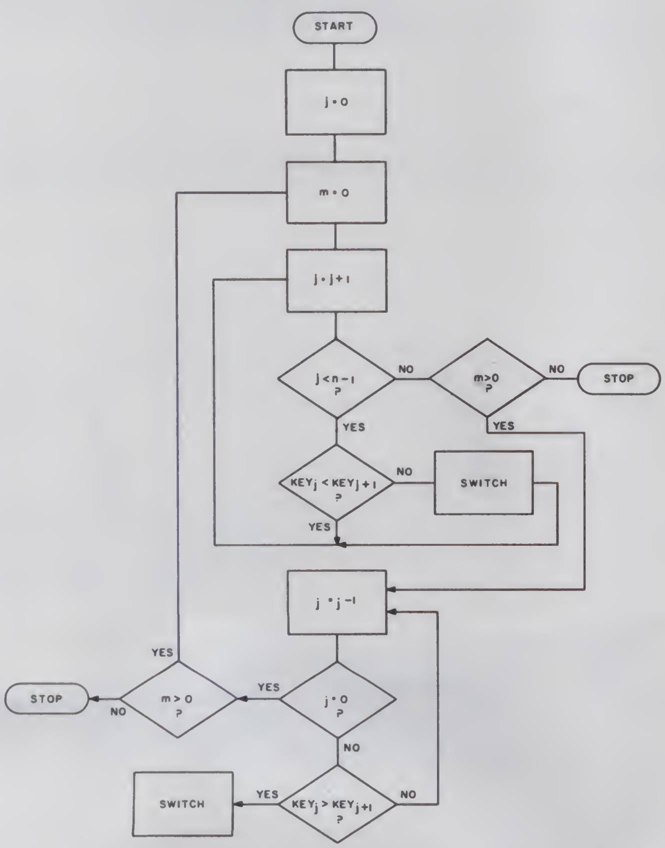
The bubble sort compares each adjoining pair of keys in the list. If they are in the proper order (i.e., the first key is less than the second key), you compare the next pair of keys in the list (this second pair of keys would include the second key of the previous pair as the first key of this pair).

On the other hand, if they are not in the correct order, you exchange the two keys so that they will be. This is done throughout the entire list. Each time you make a pass, you keep track of the number of exchanges you made. When you find you have made no exchanges, the entire file is sorted. You move the keys to their proper position within the list (as the key is repositioned, the rest of the record is, of course, also repositioned).

The bubble sort is easy to program and understand. Also, the amount of primary storage required is a function of the length of the list and the size of the records. The amount of time used for the sort depends on the length and original order of the list.

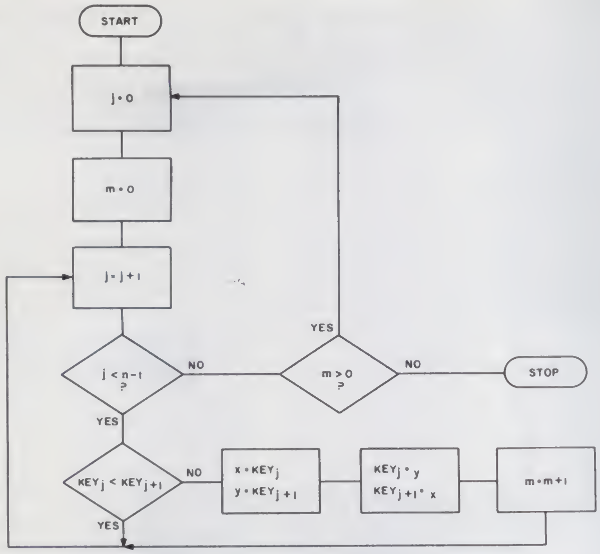
The main problem with the bubble sort is that if you have a list of n items—all in the proper order except for the key of the smallest value, which is at the very end of the list—it will require n passes to position the key at the beginning of the list where it belongs. This is because the key is only moved forward one place for each pass of the list, since the sort works from left to right.

This problem can be eliminated if you make a bidirectional pass. This method goes from left to right on its first pass, just like the single-directional pass, but then makes a pass from right to left on its return. This



Flowchart for one-way bubble sort starts with element counter j and exchange counter m at zero. As adjacent pairs of elements are compared, j is incremented. At the end of each pass, (j>n-1)—the exchange counter m—is checked to determine if an exchange had been made on this pass (m>0). If not, the sort is complete, and the job is terminated; otherwise, control is transferred to the beginning of the flow for another iteration.

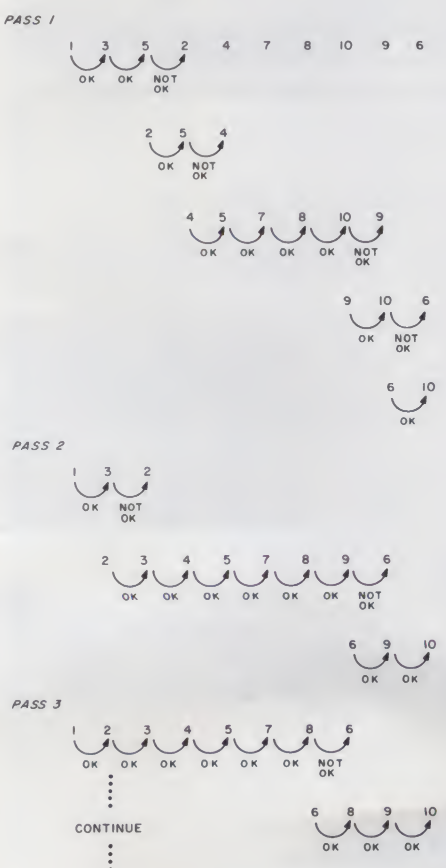
Address correspondence to W. A. Harrison, PO Box 1073, Rolla, MO 64501, and S. H. Sachs, 11 Peabody Ter., #1601, Cambridge, MA 02138.



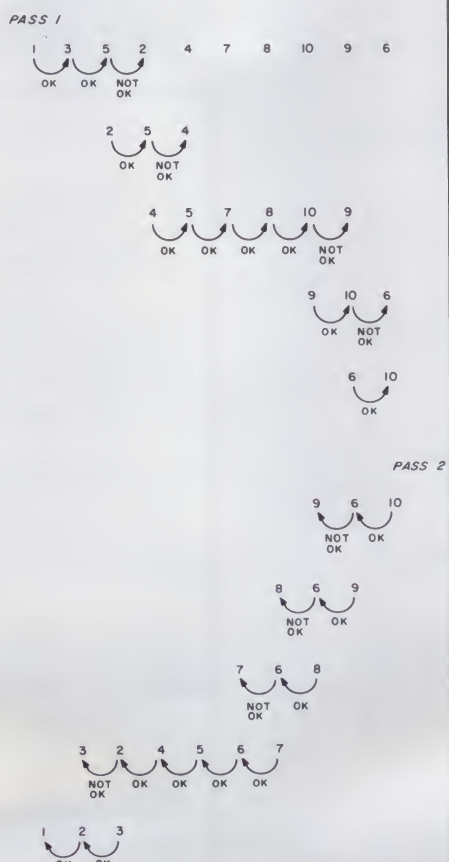
Two-way bubble sort resembles the one-way bubble sort, but instead of beginning the iteration over at the end of one pass, the counter is deincremented, and the sort operation takes place in the reverse direction. This method, as the one-way method, checks to determine if the list is sorted by checking the switch counter.

allows the out-of-order key discussed above to be properly positioned after only two passes.

With this modification, the bubble sort is an efficient, yet simple method of sorting an unordered list.■



One-way bubble sort is easy to understand and efficient, but if an unordered element is isolated in the middle of the list, a number of passes may be required to sort.



Two-way bubble sort proves to be quicker and more efficient than one-way bubble sort, especially when the unsorted element is isolated in the middle or far end of the list.

✓Reader Service—see page 210

Pet • Atari • Apple • OSI Kim • Sym • AIM

You Ought To Be Reading

COMPUTE!

The 6502 Resource And Applications Magazine

Here's what some of our recent readers say about us:

Ontario, Canada:
... Thank you for providing us beginners in the field with what is an understandable, usable and enjoyable magazine. My Pet (and I) await each issue with bated memories!!!...

Victoria, BC:
... My vote for best magazine of the decade is: **COMPUTE!** Keep up the good work...

Akron, Ohio:
... The January Issue was my first, and my subscription order is in the mail...also I can't wait to see reprints of earlier Atari articles.

Eielson AFB, Alaska:
... My back issues of **COMPUTE!** are my best references...

Taos, New Mexico:
... You are a bright spot in the lack of data on the OSI, keep it up...

Moose Jaw, Sask.:
... Excellent magazine and getting better... Don't forget that most of us are not experts...

Salinas, California:
... The first issue of my new subscription is worth the one year subscription price!

Waterford, Ohio:
... No doubt about it, **COMPUTE!** gets better with each issue. In its short lifetime it shows the growth and improvement I would have expected it to require 4 or 5 years to achieve.

Shouldn't you be reading COMPUTE! ?

COMPUTE! Magazine
P.O. Box 5406
Greensboro, NC 27403
(919) 275-9809

12-Issue subscription prices: US $16; Canada $18.00 (US funds); Surface rates, elsewhere in the world $20.00 (US funds)
Master Charge/Visa accepted. $1.00 billing fee. Sample issue, $2.50

APPLE II+ 48K RAM $1199.00

ATARI 400 8K RAM $399.00

APPLE PRODUCTS	LIST	OUR PRICE
Apple II + 48K	$1530	$1199.
Apple Disk w/cont	$645	$549.
Apple Disk w/o cont	$525	$469.
Apple Pascal System	$495	$399.
Apple Integer Card	$200	$159.
Apple Centronics Card	$225	$189.
Apple H.S. Serial Card	$195	$159.
Apple Silentype Printer	$595	$529.
Apple Fortran	$200	$169.
Apple Pilot	$150	$125.
Apple Writer	$75	$60.
DOS Tool Kit	$75	$60.
Apple Plot	$70	$58.
Microsoft 280 Softcard	$349	$299.
Microsoft 16K Ram Board	$195	$179.
Microsoft Fortran	$195	$179.
Buy All 3 Microsoft Prods	$739	$629.

ATARI PRODUCTS	LIST	OUR PRICE
Atari 400 8K	$499.95	$399.
Atari 400 16K	$599.95	$449.
Atari 800 16K	$1080.00	$799.
Atari 410 Recorder	$89.95	$69.
Atari 810 Disk Drive	$599.95	$499.
Atari 850 Interface	$219.95	$189.
Ram Cram 32K Board	$399.00	$279.
Atari 16 K Ram	$199.95	$159.

ATARI 400 8K UPGRADE KIT All you need to upgrade your 8 K 400 to a 16K 400 **$49.95**

SPECIALS

The Store Manager	$185.00
Visicalc	$119.00
Desktop Plan	$75.00
DB Master	$159.00

APPLE III 128K RAM-SANYO 12 in. MONITOR And Information Analyst Package **$3995**

CLOSE OUTS

Anadex 8000AP Printer	$750.
CCS 7424A Clock/Calendar	$99.
Apple The Cashier	$175.

TEC FP1500-25 Letter Quality Printer	$1595.00
IDS Paper Tiger 460-G Printer	$1129.00
Okidata Microline 80 Printer	$495.00

LJK Woodprocessing Software for the Atari 800 $149.00

16K RAM 200NS Set of 8 4116 **$29.95**

Sanyo 9 in B & W Monitor	$179.00
Sanyo 12" B & W Monitor	$249.00
NEC 12" Green Phosper Monitor	$259.00
CCS 7710A Serial Interface	$149.00
CCS 7428A Parallel Card	$109.00

Hayes Micromoden II	$319.00
MNT Comp AID & DIA Board	$299.00
MNT Comp Romplus Board	$179.00
MNT Comp Introl X/10 Sys	$269.00

↙130

master charge THE INTERBANK CARD

OLENSKY BROS., INC.
COMPUTER SALES DIVISION
3763 AIRPORT BLVD. • MOBILE, ALA. 36608
(205) 344-7448 • CST 9-6

VISA

$ $ $ $ NEW AT LEVEL IV $ $ $ $

QUICK FIX-THIS SUPER UTILITY WILL REPAIR MOST OF YOUR BLOWN DISK PROBLEMS, GAT TABLE, DIRECTORY, FORMAT WITHOUT ERASE AND MORE. JUST $34.95 ON DISK

SPOOLER-DON'T CONFUSE THIS ONE WITH THE DISK SPOOLERS THAT OTHERS OFFER, THIS ONE SPOOLS TO RAM, FORMATS AND PAGENATES, MACHINE LANGUAGE, RETURNS USE OF THE CPU WHILE PRINTING CONTINUES. JUST $29.95 ON DISK

ULTRA DOS-THE MOST POWERFUL SYSTEM AVAILABLE FOR THE S-80'S, INCLUDES A COMPLETE UTILITIES PACKAGE. ASK FOR AN OVERVIEW. $119.95 ON DISK (DON'T MISS THE INTRO SPECIAL PRICE, IT ENDS SOON!!!) INTRO $89.95

MODEL III GAMES AND UTILITIES, WE HAVE THE MOST COMPLETE SELECTION IN THE INDUSTRY, OTHELLO, YATZEE, GUNNERS, AND MANY OTHERS, YES! WE HAVE THE SUPER GRAPHICS ARCADE GAMES. FROM $9.95

LEVEL IV PRODUCTS INC.
32238 SCHOOLCRAFT F-4
LIVONIA, MICH. 48150
MICH: (313) 525-6200
OTHERS 800-521-3305

LEVEL IV PRODUCTS

↙23

↙315

The Atari 800 shows promise as an exciting micro for home and educational applications.

A Bright Star Comes into Focus

By William L. Colsher

In the short time that I've had my Atari 800, I've accumulated the usual assortment of nitpicks. I've also come away feeling pretty good about the system. It should give Apple and Radio Shack a run for their money.

What You Get

Once you've paid your money, what do you take home? With the Atari 800 you get an 8K, 6502-based home computer that includes a 10K operating system ROM cartridge, a BASIC ROM cartridge and an Educational System ROM cartridge. The price of the 800 also includes the Atari 410 Program Recorder (extra with the model 400 computer). For documentation you have the 332-page book *Atari BASIC* plus another 30-odd pages of operator's manuals. A cassette titled "An Invitation to Programming" wasn't quite ready when I got my system but should be packed with each computer by the time you read this.

The basic Atari 800 costs about the same as a 16K Apple II or a 16K Level II TRS-80. (These are all list prices—you may be able to get a better deal elsewhere.) You might think that Atari's memory prices are way out of line. But you're getting more than the eight simple chips an Apple or TRS-80 uses. If you compare the prices of S-100 16K memory boards, Atari's prices come into better focus. For example, a Godbout 16K Econoram IV costs $289 in "unkit" form (you plug in the chips). (See the hardware photos.)

Disk drive and printer prices are *way* out of line—sort of. *Everything* that Atari makes to connect with their computer (except joysticks) interfaces by way of a *serial* connector on the side of the console (look at the hardware photos again). You start off with the 410 tape recorder and daisy chain your printer, disks and so on. This means that every peripheral, even a printer that costs $250 somewhere else, has to be smart. The processor and support circuitry end up costing as much as the peripheral itself.

The documentation is rather strange. That 332-page book is not really a reference manual. In fact, it is a programmed instruction manual similar to the Level I Radio Shack manual. It can be used as a reference but it is far, far from complete. (To Atari's credit there is another manual —the *Atari BASIC Reference Manual*

Photo 1. The Atari 800. Note the game controller ports under the keyboard, Atari 410 Program Recorder to the far right. Program on screen is from Sears Program Library.

Address correspondence to William L. Colsher, 4328 Nutmeg Lane, Apt. 111, Lisle, IL 60532.

Photo 2. On the right side of the console are (from left to right): color monitor output, peripheral port (everything Atari makes goes through this connector), TV channel selector, power switch, and power cable. Like the TRS-80, the Atari computers use an external transformer.

—that you can get by sending in your warranty card. Mine's been in a month and I still haven't heard anything.)

The version of BASIC supplied is pretty standard. It includes the usual features such as floating point math and strings, plus new keywords for the special Atari features. There are five keywords for graphics, four for the hand controllers (joysticks or paddles), the SOUND keyword to control four-voice music or sound effects and a few more undocumented ones that I haven't been able to figure out.

While I'm on the subject of undocumented features, look at Table 2. By snooping around a little and making some educated guesses I came up with a list of keywords and functions that aren't mentioned in *Atari BASIC*. (If you'd like to snoop around a little try out Program 1. It's a refined version of the one I used to explore my new 800's ROMs.)

I only have one other complaint, about the keyboard. When I started to use my system, the keyboard had a familiar feel. So I pried off a keycap. Sure enough: little gold fingers. The Atari design seems to be a bit different than Radio Shack's, though. The fingers come together by themselves when you take off the keycap. Radio Shack's are pushed together. In addition, you get audio feedback with each keypress. (A cute little beep that slowly turns into a perfectly hideous little beep as you type in long programs.) I haven't had any trouble with keybounce (yet) so hopefully Atari learned the Radio Shack keyboard lesson and did it right the first time.

After all this complaining I should mention some of the little things Atari put in that nobody else seems to have thought of. With the standard display you get 24 lines of 38 characters. If you have a monitor or a perfectly adjusted TV set you can change the width to 40 characters, or lower it to 36 characters if your TV is poorly

Item	Price
Atari 800 Computer	$999.00
Atari 820 Printer	$599.00
Atari 810 Disk	$749.00
8K Memory	$124.00
16K Memory	$249.00
Joysticks	$19.99/pair
Game Paddles	$19.99/pair

Table 1. List prices.

Photo 3. Cover off—two ROM cartridge slots in the front section and four slots for ROM and RAM expansion to the rear. Note the heavy metal shielding. Now you know how Atari passed the FCC tests.

✓Reader Service—see page 210

Photo 4. A RAM card and the two ROM cartridges that come with the system.

adjusted. Just a couple of pokes do it. The system reset key is well off to the side and doesn't blow your program when you hit it. You have to type NEW to do that. There are probably other things that are so obvious I haven't noticed them.

Atari BASIC

Atari BASIC is a good, though smallish, version of the language. It has enough nice features to make up for the usual missing ones. Line renumbering is absent, of course (*why* can't they put it in?). More seriously, there is no automatic line numbering; you have to type each line number as you go along. To make up for these lacks is the Editor.

The Atari has a full screen editor that makes changing programs almost fun. A set of cursor control keys and some special command keys let you move around on the screen at will. You can insert and delete lines, erase extraneous characters and so on. You don't have to worry about extra blanks taking up room in your system either—if you leave a few here and there, BASIC pulls them out. (It will also insert spaces between keywords to improve readability when you LIST a program.)

You can use abbreviations with this BASIC (which makes up for having to type in the line numbers). They aren't documented but since a few are mentioned in the *Atari BASIC* book I thought I'd look for some more. Table 3 is the list of keyword abbreviations I've been able to find.

ADR	
CLOG	Log. base 10
CLOSE	
CLR	Frees all dimensioned space and sets scalars to zero
COM	Common area to share between programs? (ala FORTRAN COMMON)
CONT	Continue after a "STOP"
DEG	Tell next trig. function to expect degrees.
DOS	
ENTER	
GET	
LOAD	
LOCATE	
LPRINT	Obvious
NOT	Logical operator—n=0: NOT n=1, n≠0: NOT n=0
NOTE	
OPEN	
PADDLE	Control game paddle
PEEK	The usual
POINT	
POKE	The usual
POP	
POSITION	Place the cursor at a specific position—ex. POSITION 10,5 puts cursor at location 10 of line 5
PTRIG	Game paddle trigger
PUT	
RAD	Tell next trig. function to expect radians—default
RAND	
SAVE	
STATUS	
STICK	Control joystick
STRIG	Joystick trigger
TRAP	
USR	User machine language functions?
XIO	Direct Input/Output control?

Table 2. Keywords and functions they didn't tell us about.

Skyles Electric Works

66

BASIC Programmer's, Toolkit™, Disk-O-Pro™, Command-O™

For CBM™ Owners Who Want More Fun And Fewer Errors with Their Programs

Here are nineteen commands you'll need, on a single chip you can install in two minutes without tools, **on any CBM or CMB system.** 4KB of ROM firmware on each chip with a collection of machine language programs available from the time you turn on your PET to the time you shut it off.

For CBM 8016 and 8032; BASIC 4.0

BASIC Programmers Command-O™

AUTOed **DUMP**ed **DELETE**ed **FIND**ed (improved) **HELP**ed **KILL**ed **OFF**ed **TRACE**ed (improved) **RENUMBER**ed (improved) **INITIALIZE**BS **MERGE**BS **MOVE**BS **EXECUTE**BS **SCROLL**ed **OUT**ed **SET**ed **SEND**BS **PRINT USING**BS **BEEP**BS

```
  100 GOSUB 180
  105 PRINT USING C$, A, B$
  130 INPUT "TIME", D$
  131 INPUT "DAY", E$
  160 IFB<  >C THEN 105
  180 FOR X=IT09
  183 PRINT Y(X):NEXT
  184 RETURN
  200 I=X/19
READY

RENUMBER 110, 10, 105-184

READY
LIST
  100 GOSUB 150
  110 PRINT USING C$, A, B$
  120 INPUT "TIME", D$
  130 INPUT "DAY", E$
  140 IFB< >C THEN 110
  150 FOR X=IT09
  160 PRINT Y(X):NEXT
  170 RETURN
  200 I=X/19
READY
```

```
MERGE D1 "BUY NOW*"

SEARCHING FOR BUY NOW*
LOADING

READY
RENUMBER 100, 10

READY
FIND B$
110 PRINT USING A$, B$, B$+C$+D$
280 B$="NOW IS THE TIME"

READY
```

```
 580 BA=BA 1
 590 RA=123*5X/92+BA*10
 600 IF BA=143 THEN 580
 610 RETURN
 620 C$="PROFIT $#, ###.## DAILY"
 630 PRINT USING C$, PI
 640 D$="LOSS  $#, ###.## DAILY"
 650 PRINT USING D$, LI

RUN

PROFIT $1, 238.61 DAILY
LOSS   $     0.00 DAILY

READY
```

```
  180 FOR A=4096T08191:POKEA, B:B=B+1:PRIN
T B:IFB 255 THEN B=B-255:PRINT B

TRACE
```

PRICES:

BASIC Programmers **Toolkit™** (chip only)	$40.00
BASIC Programmers **Disk-O-Pro™** (chip only)	$75.00
BASIC Programmers **Command-O™** (chip only)	$75.00
Interface boards (needed sometimes)	$20.00-$50.00
Instruction Manual (with redeemable $5.00 coupon)	$5.00

Shipping and handling $2.50 USA/Canada, $10.00 Europe/Asia

California residents please add 6% or 6-1/2% sales tax as required

*Reserve your **Disk-O-Pro, Command-O** today*

Toolkit™ immediate delivery, Disk-O-Pro delivery in December, Command-O delivery in January

VISA, MASTERCHARGE ORDERS CALL (800) 538-3083 (except California residents)

CALIFORNIA ORDERS PLEASE CALL (408) 257-9140

Skyles Electric Works

231 E South Whisman Road
Mountain View, CA 94041
(415) 965-1735

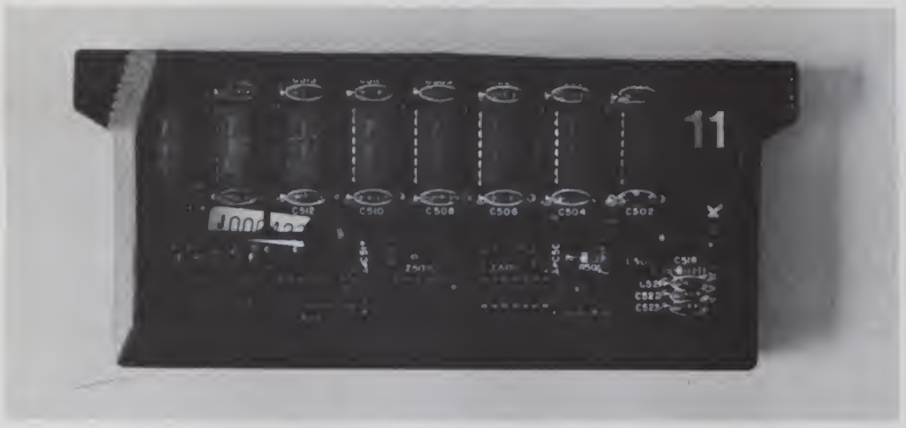


Photo 5. Inside a RAM card. Just your regular 16K dynamics.

If you use a keyword or abbreviation when typing in a program, BASIC translates it to an internal form. When you LIST the program you get the *whole* keyword rather than the short form. This is another aid for getting readable programs from what you type in.

One of the very nicest features of this version of BASIC is one I've never seen before. If you make a mistake while typing in a BASIC program, the computer will tell you right away with the cursor at the point the error was detected. Since you have a terrific screen editor you can just go back to your original statement, make the changes and continue.

As you may know from reading the advertisements, this computer has nice graphics. Here are some you may not know about. The keyboard is full upper and lowercase. (In fact, the keyboard looks a whole lot like a standard *terminal* keyboard. It even has square brackets. I think somebody had timesharing in mind when this machine was designed.) Then there are the 29 keyboard graphics characters that you can use with

Keyword	Abbreviation
COLOR	C.
CSAVE	CS.
DATA	D.
DRAWTO	DR.
FOR	F.
GOSUB	GOS.
GOTO	G.
INPUT	I.
LIST	L.
LPRINT	LP.
NEXT	N.
PLOT	PL.
PRINT	?
REM	R.
RESTORE	RES.
RETURN	RET.
SETCOLOR	SET.
SOUND	SO.

Table 3. Keyword abbreviations.

Hidden Features Exposed

By Jerry Blank

The Atari has gotten far less recognition than it deserves. It has the program-editing features of expensive terminals, eight built-in A/D converters, selectable channel 2 or 3 rf output and hi-res graphics capabilities to match those of the Apple II. Unlike the Apple, however, Atari has a very decent and thorough BASIC, with 10K operating system support and disk commands to boot. Atari BASIC's one minor flaw, perhaps, is the absence of user-defined functions and string arrays.

But Atari users have one major complaint—the *Atari Reference Manual* is vague about the use of the 410 Program Recorder for data table storage.

Address correspondence to Jerry Blank, RR1 N, Box 71B, Williston, ND 58801.

Software Modifications

Atari users are probably aware that the I/O commands PUT and GET are useful in screen graphics. In addition, these commands can be used to access other parts of the computer. In the statement

```
OPEN #2,12,0,"S"
```

I/O port 2 is reserved for the screen ("S"). The letter within quotes can be changed, defining a different part of the computer. In the case of the Program Recorder, this letter is "C." Therefore, access to the cassette can be obtained with the following statement:

```
OPEN #N,12,0,"C"
```

N can be any I/O port from 2 through 6, provided the user has not reserved other ports with previous OPEN statements. A typical routine to save an array on tape looks like this:

```
10 OPEN #2,8,0,"C"
20 PRINT "Press any key to start tape."
30 FOR I= 1 TO 100
40 PUT #2, A(I)
50 NEXT I: CLOSE #2
60 ?"Data saved."
```

To recover the data, use the same routine but change the following lines:

```
10 OPEN #2,4,0,"C"
40 GET #2,X: A(I)=X
60 PRINT "Data loaded."
```

The cassette motor *will operate* automatically, but make sure the play and record buttons are pressed as necessary. Also, you might need to advance the tape past the leader at the beginning of the cassette.

Along with cassette file saving, there are many other interesting and useful POKE locations. For example, if you have tried printing some of the Atari graphics characters, you will know that some characters must be preceded with "ESC" to be printed, which can be downright inconvenient. This little problem can be solved by poking 1 into memory location 766. But doing this will disable all the fancy editing functions, so

```
10 PRINT "42159 TO 42494"
20 FOR I=42159 TO 42494
30 PRINT CHR$(PEEK(I));
40 NEXT I
50 PRINT :PRINT "42986 TO 43133"
60 FOR I=42986 TO 43133
70 PRINT CHR$(PEEK(I));
80 NEXT I
90 END
```

Program 1. This particular "snooper" displays the Atari 800's BASIC keyword tables. This is a good way to check up the completeness of the documentation supplied with a computer. To get a polished snooper like this you have to start with location zero and just let it run until you see something interesting on the screen. Then you fine-tune to display just what you want to look at.

PRINT statements. As you can see from Photo 6 all this is available in reverse video as well.

Atari BASIC documents six graphics modes. As you can see from Table 4 I've been able to figure out nine. I can understand not putting Mode 8 in a general-purpose book sent out with 8K systems, since that mode requires more than 8K for the screen memory. (The Atari 400 computer can only *have* 8K so it can never use Mode 8.)

Atari BASIC states that a "complete color graphics program must have the three statements: GRAPHICS, COLOR and SETCOLOR." Well, GRAPHICS is used to set the mode. That's pretty easy to understand. COLOR is used to turn on the "color machine," though it is never really explained. I've discovered some other uses for it as well. SETCOLOR is never fully explained. It is the most complicated of the three statements and could use better coverage. Frankly, I don't understand it at all. There isn't much information in the book and the instruction seems to work differently with different graphics modes. Program 2 is a simple demonstration of the Atari's color graphics. Its output is shown in Photo 7. What you can't see in the picture is that after each execution of the SETCOLOR statement *all* the points that have been plotted change to the new color.

The secret to getting more than one color on the screen at once seems to lie in the use of the COLOR statement. But it's not mentioned exactly how to do this. Here's what seems to happen:

- With COLOR 0 all points plotted are the same as the background color (i.e., they're invisible—an erase function).
- With COLOR 1 all points change color when a SETCOLOR 0,c,1 is exe-

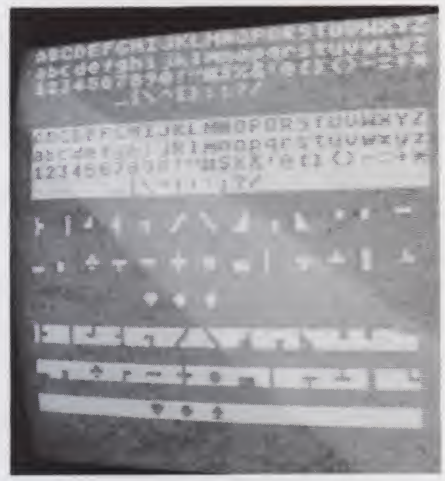


Photo 6. The Atari character set.

---

when you're through printing control characters, load location 766 with 0.

In graphics modes 1 and 2, it is only possible to display uppercase and numbers/punctuation, or lowercase and special characters—not both at one time. To obtain uppercase in these modes, poke 224 into location 756. To get lowercase characters, *poke 226*.

In some Atari arcade games using the same graphics chip as the 400/800, the display is a mirror reflection of the CRT—a clever space-saving technique. To make a right-reading image in the mirror, the CRT must display a reverse mirror image. You can get this mirrored video effect by poking 4 into location 755. To restore normal video, poke 2 into location 755.

In some programs, particularly games, it may be advisable to turn off the cursor, by changing the contents of memory location 752 from 0 to 1.

If you wish to start the cassette motor remotely, poke 52 into location 54018. To turn the motor off, poke 60 into the same location.

One interesting feature of the Atari is the real-time clock, located at memory addresses 18, 19 and 20. This clock is updated every 1/60 of a second. Here is a simple 24-hour clock program that implements the real-time feature:

```
10 GRAPHICS 0: PRINT "Set hours/ minutes
   seconds";: INPUT H,M,S
20 A=INT(((H*3600) + (M*60) +S)*60):
   B=INT(A/65536):C=INT((A-(B*65536))
   /256): D=A-(B*65536)-(C*256)
30 GRAPHICS 0: POKE 18, B: POKE 19,C:
   POKE 20,D
40 A=INT(((PEEK(18)*65536) +
   ((PEEK(19)*256 + PEEK(20)))/60):
   B=INT(A/3600):C=INT((A-(B*3600)):
   D=A-(B*3600)-(C*60)
50 POSITION 2,0: PRINT B;":";C;":";D;" "
60 GOTO 40
```

To modify this program to reset hours at midnight or to have a selectable 12/24-hour mode, only minor changes are needed.

Hardware Modification

Owners of 8K Atari 400s soon find out that 8K is not nearly enough user memory, especially when using 160 × 80 four-color graphics. In fact, it is impossible to use the high-resolution 320 × 160 graphics with a minimal 8K system. Unfortunately, the Atari 400 has no provision for direct memory expansion. The 8K memory card may be replaced with a 16K board at an approved regional service center, but this is time-consuming and costly.

You can upgrade your Atari 400 to 16K using this method:

- Disassemble an Atari CX853 16K memory cartridge.
- Remove the PC board inside and discard the plastic case and screws.
- Remove the Atari 400's cover and metal rf interference shield.
- There are two printed circuit cards plugged into the main board just behind the ROM cartridge slot. The front-most board is the 8K memory board, and is identical with its 16K replacement. Simply replace the board with the one taken from the CX853 cartridge, reassemble the computer and turn it on.

There should now be 13,326 bytes free. Keep in mind that servicing an Atari computer yourself is a violation of the 90-day warranty.

It seemed to me that the 16K board was merely an 8K board with 16K chips, and the two discrete resistors near the edge connector clipped off. I didn't try to directly modify my 8K board, though; it is easier, although more expensive, to plug in a new board.■

cuted (c is the color and 1 is luminance).

- With COLOR 2 all points change color when a SETCOLOR 1,c,1 is executed.
- With COLOR 3 all points change color when a SETCOLOR 2,c,1 is executed.

So it's actually possible to have three colors, plus the background color, on the screen at once.

If you take a look at Table 4 again you'll notice that graphics modes 1 and 2 display *letters*. To use this feature you need the COLOR and PLOT statements. In COLOR you use the ASCII value of the character you want to use and then PLOT at the position you want it to appear. With a little fooling around it turns out you can get these letters in four different colors as well.

In addition to the PLOT command, Atari has supplied a command called DRAWTO. It does exactly what you would expect it to: it draws a line from the point you're at to the one you specify. Of course, you have to use a PLOT or previous DRAWTO so the computer has a place to start. (Actually, the computer will try to draw a line for you but the results will be unpredictable if you haven't given it a starting point.)

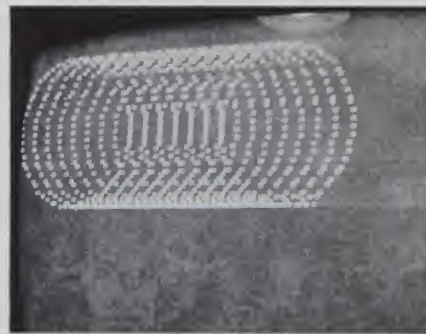


Photo 7. Mode 7 graphics. The highest resolution available on a minimum system.

A couple of years ago it was a big deal to toggle an output bit to make music. Then came some simple boards that improved on that technique a little. Now we have a four-voice synthesizer in a $1000 personal computer.

The tone quality is excellent. There are 256 possible notes, which range from just barely audible (inaudible to some people) at the high end to about an octave below middle C.

Using the SOUND feature is also easy. *Atari BASIC* contains a table of "computer notes" (0 to 256) and "musical notes" (D#, C, F, etc.). You don't have to play just music, of course. By playing with the *tone* parameter, you can get some pretty strange effects. Interestingly, only the even tones (0,2,4,8, etc.) produce any sound.

Obviously, the possibilities here are pretty wide. I've come up with a few ideas for strange, science-fiction sounds in Program 3. You can play regular music with chords and a melody, since once you turn on a note with the SOUND statement *it doesn't* go off until you turn it off.

Games improved when we got graphics with our computers. I think they'll improve some more now that sound is easy and standard. The edu-

```
10   GRAPHICS 7
20   COLOR 1
30   FOR C=0 TO 15
40   SETCOLOR 0,c,10
50   GOSUB 1000
60   NEXT C
70   STOP
1000 FOR TH=0 TO 6.28 STEP .1
1010 X=SIN(TH)*20+20+C*5
1020 Y=COS(TH)*20+20
1030 PLOT X,Y
1040 NEXT TH
1050 RETURN
```

Program 2. The program to produce Photo 7.

Mode	Overhead	Resolution or Result
0	3058	Regular screen
1	2740	Double wide letters and 4 line window at bottom
2	2490	Double wide & double high letters and 4 line window
3	2500	39×20 and 4 line window.
4	2760	79×40 .. ..
5	3240	79×40 .. ..
6	4240	158×80 .. ..
7	6256	158×80 .. ..
8	10178	320×192 .. ..

Table 4. Graphic mode overhead.

```
10   REM ***LISTEN TO A SINE WAVE
20   FOR TH=0 TO 3.1415 STEP .1
30   N=125*ABS(SIN(TH))
40   SOUND 0,N,10,8
50   NEXT TH
100  REM ***PHASER FIRE
110  FOR N=20 TO 40
120  SOUND 0,N,10,8
130  NEXT N
200  REM ***BATTLE CRUISER
     ZOOOMING BY
210  FOR V=0 TO 15
220  SOUND 0,25,16,V
230  FOR I=1 TO 20:NEXT I
240  NEXT V
250  FOR V=15 TO 0 STEP -1
260  SOUND 0,25,16,V
270  FOR I=1 TO 20:NEXT I
280  NEXT V
999  END
```

Program 3. Some interesting sounds.

Since this article was written a number of things I mentioned are no longer true and some of my predictions have proven accurate. First: the Atari *BASIC Reference Manual* is now packed with each system. It is an excellent manual for the experienced programmer and the beginner will be able to use it as soon as he finishes the *Atari BASIC* book. Prices have gone up. The Atari 800 now retails for about $1080, but it includes 16K of memory rather than 8K and the 410 tape recorder is extra.

A number of companies have entered the add-on market and at least one offers a 16K add-on for $119.50 and a 32K (!) board for $320. The possibilities opened by the 32K board are quite interesting since it leaves one memory slot completely free. There are also several printer interfaces available that make use of the game ports, and software is becoming increasingly easy to find.

cational possibilities are interesting, too. With an accurate note source built into the computer, instruction in music theory and even elementary instrumental training are possible. (As a matter of fact, I'm working on an elementary guitar course right now.)

Conclusions

Would I still buy an Atari? Yes, I think so. My major complaint was about the documentation—I think that Atari went a litte too far in supporting the beginner at the expense of the experienced programmer. I am concerned about possible problems when running all peripherals daisy-chained. For example I have read that the disk has a transfer rate of 48,000 bps. My MPI disk drive manual says the drive is capable of handling up to 250,000 bps. That doesn't seem like very efficient use of a rather expensive peripheral.

I have faith in this industry, though. If the Atari becomes popular enough people will come out with add-ons that use the memory board slots rather than the peripheral port on the side of the machine. I anticipate that with the great potential of this machine, it won't be long before we see those add-ons.■

MCP

COMPARE THE FEATURES, COMPARE THE PRICE. ORIGINALLY DESIGNED FOR OEM APPLICATIONS. MCP SYSTEMS COMPONENTS OFFER HIGH RELIABILITY AND PERFORMANCE AT AN AFFORDABLE PRICE.
ALL ARE CP/M AND IEEE/S100 COMPATIBLE.

THE MCP/SCC Z80 CPU CARD FEATURES:
TWO SERIAL PORTS WITH RS232 DRIVE AND HANDSHAKING. BAUD RATES FROM 50 TO 19.2K. AN 8255 PROGRAMMABLE PARALLEL INTERFACE. 1K ONBOARD STATIC RAM AND 1K EPROM. WITH MONITOR PROGRAM FOR SERIAL TERMINAL INCLUDED. 2K 2716 EPROM OPTIONAL. ONBOARD RAM/EPROM OPERATE IN SHADOW MODE TO ALLOW 64K ADDRESSING. OPERATES AT 2 OR 4 MHZ.
ASSEMBLED AND TESTED LIST $349.95 **SALE PRICE $299.95**

THE MCP/FDC DOUBLE DENSITY DISK CONTROLLER:
UTILIZES THE WD1793 DOUBLE DENSITY F/D CONTROLLER CHIP AND THE WD1691 FLOPPY SUPPORT LOGIC CHIP. THIS CHIP SET OFFERS SUPERB PERFORMANCE AND RELIABILITY DUE TO ENHANCED DATA RECOVERY AND WRITE PRECOMP SUPPORT. THE MCP/FDC WILL SUPPORT UP TO FOUR 8" SINGLE OR DOUBLE SIDED SHUGART OR SHUGART COMPATIBLE DRIVES. THE MCP/FDC OPERATES IN SINGLE OR DOUBLE DENSITY MODE WITH 4 MHZ SYSTEMS.
ASSEMBLED AND TESTED LIST $359.95 **SALE PRICE $329.95**

THE MCP/VDB 80X24 VIDEO BOARD FEATURES:
A 7X9 CHARACTER MATRIX WITH LOWERCASE DESCENDERS. MEMORY MAPPED WITH 2K ONBOARD RAM. KEYBOARD INPUT PORT. PROGRAMMABLE FORMAT. HARDWARE SCROLL. NON-DESTRUCTIVE CURSOR WITH 2 BLINK RATES. AND REVERSE VIDEO. ASSEMBLED AND TESTED LIST $289.95 **SALE PRICE $249.95**
MCP/VDB BARE BOARD AND DOCUMENTATION. $60.00

COMPLETE MCP 80/100 MAINFRAMES: DUAL 8" SINGLE OR DOUBLE SIDED SHUGART DISK. (CALL FOR LOW PRICE)

SOFTWARE:
- • TCS ACCOUNTING PACKAGE: $249.95 (REQUIRES CP/M AND MICROSOFT BASIC)
- • MAGIC WAND WORD PROCESSING PACKAGE: $339.99
- • TICS (TOTAL INVENTORY CONTROL SYSTEM) BY J&S SOFTWARE. (CALL FOR PRICE)
- • HUFF & PUFF (BIT COMPRESSION/EXPANSION) BY J&S SOFTWARE $50.00

MCP ✓316
COMPUTER PRODUCTS

SUITE 444D
6992 EL CAMINO REAL
CARLSBAD, CA. 92008
PHONE (714) 438-3270

is HARD COPY STORAGE a problem?

KILOBAUD MICROCOMPUTING, as thick as it is, is more like a floppy when it comes to standing on the bookshelf. Try the **KILOBAUD MICROCOMPUTING** Library Shelf Boxes . . . sturdy corrugated white dirt-resistant cardboard boxes which will keep them from flopping around. We have self-sticking labels for the boxes, too, not only for **KILOBAUD MICROCOMPUTING,** but also for **73 Magazine, 80 MICROCOMPUTING** . . . and for **CQ, QST, Ham Radio, Personal Computing, Radio Electronics, Interface Age,** and **Byte**. Ask for whatever stickers you want with your box order. They hold a full year of **KILOBAUD MICROCOMPUTING, 80 MICROCOMPUTING** . . . or **73 Magazine**. Your magazine library is your prime reference; keep it handy and keep it neat with these strong library shelf boxes. One box (BX-1000) is $2.00, 2-7 boxes (BX-1001) are $1.50 each, and eight or more boxes (BX-1002) are $1.25 each. Be sure to specify which labels we should send. Have your credit card handy and call our toll-free order number 800-258-5473, or use the order card in the back of the magazine and mail to:

kilobaud TM
MICROCOMPUTING
peterborough nh 03458

APPLE II TRS·80

Ⓐ QUALITY Ⓣ
VISA **DISK SOFTWARE** master charge

☐ HOME FINANCE PAK I: Entire Series $49.95 Ⓐ Ⓣ

☐ CHECK REGISTER AND BUDGET: This comprehensive CHECKING ACCOUNT MANAGEMENT SYSTEM not only keeps complete records, it also gives you the analysis and control tools you need to actively manage your account. The system provides routines for BUDGETING INCOME AND EXPENSE, AUTOMATIC CHECK SEARCH, and BANK STATEMENT RECONCILING. CRT or printer reports are produced for ACTUAL EXPENSE vs BUOGET, CHECK SEARCH DISPLAY, RECONCILIATION REPORT and CHECK REGISTER DISPLAY by month. Check entry is prompted by user-defined menus of standard purposes and recipient codes, speeding data entry and reducing disk storage and retrieval time. Six fields of data are stored for each check: amount, check no., date, purpose, recipient and TAX DEOUCTIBLE REMINOER. CHECK SEARCH routines allow searching on any of these data fields. Routines are also provided for CHECK SORT by date and check no., OATA EDITING and Report Formats. Up to 100 checks/mo. storage $39.95

☐ SAVINGS: Account management system for up to 20 separate Savings accounts. Organizes, files and displays deposits, withdrawals and interest earned for each account. Complete records shown via CRT or printer. $14.95

☐ CREOIT CARO: Get control of your credit cards with this program. Organizes, stores and displays purchases, payments and service charges for up to 20 separate cards. Use for credit cards or bank loans. CRT or printer reports $14.95

☐ UNIVERSAL COMPUTING MACHINE: $49.95 Ⓐ Ⓣ

A user programmable computing system structured around a 50 row x 50 column table. User defines row and column names and equations forming a unique computing machine. Table elements can be multiplied, divided, subtracted or added to any other element. User can define repeated functions common to a row or column greatly simplifying table setup. Hundreds of unique computing machines can be defined, used, stored and recalled, with or without old data, for later use. Excellent for sales forecasts, engineering design analysis, budgets, inventory lists, income statements, production planning, project cost estimates-in short for any planning, analysis or reporting problem that can be solved with a table. Unique curser commands allow you to move to any element, change its value and immediately see the effect on other table values. Entire table can be printed by machine pages (user-defined 3-5 columns) on a 40 column printer:

☐ COLOR CALENDAR: $29.95 Ⓐ

HI-RES color graphics display of your personal calendar. Automatic multiple entry of repetitive events. Review at a glance important dates, appointments, anniversaries, birthdays, action dates, etc. over a 1 year period. Graphic calendar marks dates. Printer and screen display a summary report by month of your full text describing each day's action item or event. Ideal for anyone with a busy calendar.

☐ BUSINESS SOFTWARE: Entire Series $159.95 Ⓐ Ⓣ

☐ MICROACCOUNTANT. The ideal accounting system for the small business. Based on classic T accounts and double-entry bookkeeping, this efficient program records and produces reports on account balances, general ledger journals, revenue and expenses Screen or 40 column printer reports. Handles up to 1000 journal entries per month up to 300 accounts. Includes a short primer in Financial Accounting. $49.95

☐ UNIVERSAL BUSINESS MACHINE: This program is designed to SIMPLIFY and SAVE TIME for the serious businessman who must periodically Analyze, Plan and Estimate. The program was created using our Universal Computing Machine and it is programmed to provide the following planning and forecasting tools

CASH FLOW ANALYSIS	PROFORMA BALANCE SHEET	SDURCE AND USE DF FUNDS
PROFORMA PROFIT & LDSS	SALES FDRECASTER	JDB CDST ESTIMATDR

Price, including a copy of the Universal Computing Machine $89 95

☐ BUSINESS CHECK REGISTER AND BUDGET: Our Check Register and Budget programs expanded to include up to 50 budgetable items and up to 400 checks per month. Includes bank statement reconciling and automatic check search (48K) $49.95

☐ ELECTRONICS SERIES: Entire Series $259.95 Ⓐ Ⓣ

☐ LOGIC SIMULATOR: SAVE TIME ANO MONEY. Simulate your digital logic circuits before you build them. CMOS, TTL, or whatever, if it's digital logic, this program can handle it. The program is an interactive, menu driven, full-fledged logic simulator capable of simulating the bit-time by bit-time response of a logic network to user-specified input patterns. It will handle up to 1000 gates, including NANOS, NORS, INverters, FLIP-FLOPS, SHIFT REGISTERS, COUNTERS and user-defined MACROS. Up to 40 user-defined, random, or binary input patterns. Simulation results displayed on CRT or printer. Accepts network descriptions from keyboard or from LOGIC DESIGNER for simulation. $159.95

☐ LOGIC OESIGNER: Interactive HI-RES Graphics program for designing digital logic systems. A menu driven series of keyboard commands allows you to draw directly on the screen up to 15 different gate types, including 10 gate shape patterns supplied with the program and 5 reserved for user specification. Standard patterns supplied are NANO, NOR, INVERTER, EX-OR, T-FLOP, JK-FLOP, D-FLOP, RS-FLOP, 4 Bit COUNTER and N-BIT SHIFT REGISTER. User interconnects gates just as you would normally draw using line graphics commands. Network descriptions for LOGIC SIMULATOR generated simultaneously with the CRT diagram being drawn. $159.95
MANUAL AND DEMO OISK. Instruction manual and demo disk illustrating capabilities of both programs $29.95

☐ MATHEMATICS SERIES: Entire Series $49.95 Ⓐ

☐ STATISTICAL ANALYSIS I: This menu driven program performs SIMPLE LINEAR REGRESSIDN analysis, determines the mean, standard deviation and plots the frequency distribution of user-supplied data sets. Printer, Oisk, I/D and edit routines included (32K min.). $19 95

☐ NUMERICAL ANALYSIS: HI-RES 2-Dimensional plot of any function. Automatic scaling. At your option, the program will plot the function, plot the INTEGRAL, plot the DERIVATIVE, determine the ROOTS, find the MAXIMA and MINIMA and list the INTEGRAL VALUE. $19.95

☐ MATRIX: A general purpose, menu driven program for determining the INVERSE and OETERMINANT of any matrix, as well as the SOLUTION to any set of SIMULTANEOUS LINEAR EQUATIONS *Disk I/O for* data save. Specify 55 eqn. set (48K) or 35 eqn. (32K) $19.95

☐ 3-O SURFACE PLOTTER: Explore the ELEGANCE and BEAUTY of MATHEMATICS by creating HI-RES PLOTS of 3-dimensional surfaces from any 3-variable equation. Oisk save and recall routines for plots Menu driven to vary surface parameters. Hidden line or transparent plotting $19 95

☐ ACTION ADVENTURE GAMES: Entire Series $29.95 Ⓐ

☐ REO BARDN: Can you outfly the RED BARDN? This fast action game simulates a machine gun DDG FIGHT between your WORLO WAR I BI-PLANE and the baron's. You can LOOP, DIVE, BANK or CLIMB in any one of 8 directions and so can the BARDN in HI-RES graphics. $14 95

☐ BATTLE DF MIDWAY: You are in command of the U.S.S. HDRNETS' DIVE-BDMBER squadron. Your targets are the Aircraft carriers, Akagi, Soryu and Kaga. You must fly your way through ZERDS and AA FIRE to make your DIVE-BOMB run. In HI-RES graphics $14.95

☐ SUB ATTACK. It's April, 1943. The enemy convoy is headed for the CDRAL SEA Your sub, the MDRAY, has just sighted the CARRIERS and BATTLESHIPS. Easy pickings. But watch out for the DESTRDYERS they're fast and deadly. In HI-RES graphics $14 95

☐ FREE CATALOG All programs are supplied on disk and run on Apple II w/Disk & Applesoft ROM Card & TRS-80 Level II and require 32K RAM unless otherwise noted Detailed instructions included. Drders shipped within 5 days. Card users include card number Add $1.50 postage and handling with each order California residents add 6½% sales tax. Foreign orders add $5.00 postage and handling.

Make checks payable to:
SPECTRUM SOFTWARE ✓306
P.O. Box 2084 142 Carlow, Sunnyvale, CA 94087
For phone orders - 408-738-4387
DEALER INQUIRIES INVITED

No list of records is too short or long for this assembly-language Shell-Metzner sort.

Sorta Super Fast

By Albert J. Marino

This program will let you sort 32K in one minute. This means you can sort 2000 names in one minute and, in fact, can sort 10,000 names in a reasonable time. The routine can easily be used with any BASIC or other language as long as it supports external (user) routines and has a POKE or similar function.

I originally wrote a machine language program that worked on the algorithm shown in Fig. 1. This routine was short and simple, required no scratch space and was about four times faster than a modified bubble sort. I have never seen this algorithm implemented by anyone else and it might be unique; it certainly is good enough for most applications.

However, when the files reached 2000 names, I found that this routine was taking two to three minutes. Since the time for the routine is approximately a function of N^2, this meant that 10,000 names would take over an hour. I decided I needed a better routine.

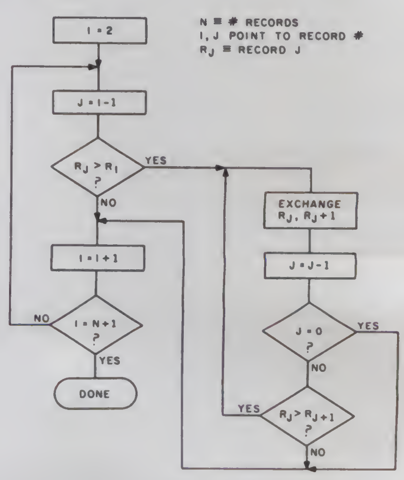


Fig. 1.

Listing 1. 8080 assembly-language Shell-Metzner sort.

```



                        ;
                        ;       General sort routine, in memory
                        ;
                        ;
                        ;
                        ;
AB00                    orig    org     0ab00h  ;change this for dif loc
AB00 =                  n1:     equ     orig
AB02 =                  m1:     equ     n1+2
AB04 =                  k1:     equ     m1+2
AB06 =                  j1:     equ     k1+2
AB08 =                  i1:     equ     j1+2
AB0A =                  m11:    equ     i1+2
AB0C =                  dj1:    equ     m11+2
AB0E =                  di1:    equ     dj1+2
                        ;
AB00                            ds      16
                        ;
AB10 2A06AB                     lhld    j1      ;get st adr
AB13 E5                         push    h       ;save
AB14 2A04AB                     lhld    k1      ;get length
AB17 E5                         push    h       ;save
AB18 AF                 div:    xra     a       ;m1=m1/2
AB19 2A02AB                     lhld    m1
AB1C 7C                         mov     a,h
AB1D 1F                         rar
AB1E 67                         mov     h,a
AB1F 7D                         mov     a,l
AB20 1F                         rar
AB21 6F                         mov     l,a
AB22 2202AB                     shld    m1      ;save new m1
                        ;
AB25 B4                         ora     h       ;check if done
AB26 C22CAB                     jnz     ndon
AB29 C1                         pop     b       ;done, return
AB2A D1                         pop     d
AB2B C9                         ret
                        ;
                                ;set k1=n1-m1
                        ;
AB2C EB                 ndon:   xchg            ;m1 to de
AB2D 2A00AB                     lhld    n1
AB30 7D                         mov     a,l
AB31 93                         sub     e
AB32 6F                         mov     l,a
AB33 7C                         mov     a,h
AB34 9A                         sbb     d
AB35 67                         mov     h,a
AB36 2204AB                     shld    k1
```

More →

Address correspondence to Albert J. Marino, Compleat Systems, 9551 Casaba Ave., Chatsworth, CA 91311.

The one I settled on was the Shell-Metzner sort shown in Fig. 2. Listing 1 shows the 8080 assembly-language program for this sort. Of course, the program will work for any 8080- or Z-80-based computer. The listing happens to be assembled at location AB00H but can be reassembled for any starting address. Note that the routine takes 203 bytes, including the 16-byte scratch area at the beginning.

Using the Routine

Once the program is in memory, it can be used by any other program, as long as it is not overwritten. To use it follow these steps:

- Put the records to be sorted into memory starting at any location and sequentially up in memory. The records must be the same length.
- Put the number of records to be sorted in location n1 (AB00H in our listing) and also in location m1 (AB02H). Note that the numbers are 16-bit (two-byte) numbers and are stored in standard reversed format with low byte first and high byte second. For example, if the number of records is 4000 (=0FA0H) then 160 (A0) is put into location AB00H and 15 (0F) is put into location AB01H.
- In like manner, put the length of each record into location k1 (AB04,5) and the starting address in location j1 (AB06,7).
- Transfer control to the program (call) at location ORG+16 (AB10H).

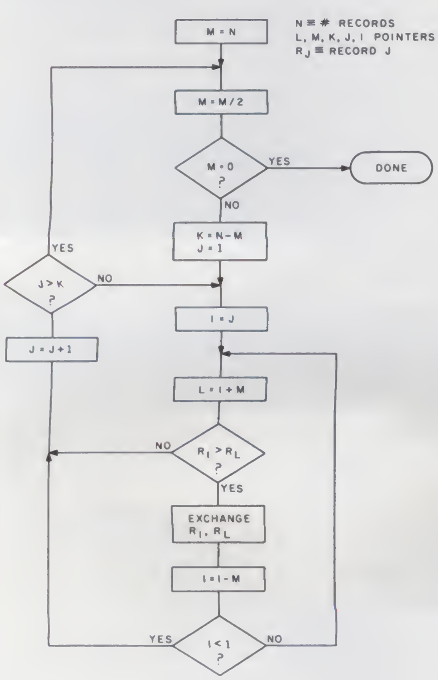


Fig. 2.

Listing 1 continued.

```
                        ;
        AB39 210100             lxi     h,1         ;set & save i=j=1
        AB3C 2206AB             shld    j1
        AB3F 2208AB             shld    i1
                        ;
                                ;calc & save addr offset = m1*l1
                        ;
        AB42 2D                 dcr     l
        AB43 C1                 pop     b           ;len of str=l1
        AB44 C5                 push    b
        AB45 19         lp1:    dad     d
        AB46 0B                 dcx     b
        AB47 78                 mov     a,b
        AB48 B1                 ora     c
        AB49 C245AB             jnz     lp1
        AB4C 220AAB             shld    ml1
                        ;
        AB4F EB                 xchg                ;calc & save d(j), d(i), d(i+m)
        AB50 C1                 pop     b
        AB51 E1                 pop     h
        AB52 E5                 push    h
        AB53 C5                 push    b
        AB54 220CAB     lp2:    shld    dj1
        AB57 220EAB             shld    di1
        AB5A EB                 xchg
        AB5B 19                 dad     d
        AB5C EB                 xchg                ;hl has d(i), de has d(i+m)
                        ;
                                ;compare strings & switch
                        ;
        AB5D C1         cp1:    pop     b           ;len of string=l1
        AB5E C5                 push    b
        AB5F 1A         lp3:    ldax    d           ;compare ea byte
        AB60 96                 sub     m
        AB61 C26FAB             jnz     neq
        AB64 23                 inx     h           ;if = compare next byte
        AB65 13                 inx     d
        AB66 0B                 dcx     b
        AB67 78                 mov     a,b
        AB68 B1                 ora     c
        AB69 C25FAB             jnz     lp3
        AB6C C3A8AB             jmp     nsw         ;if done, don't switch
                        ;
        AB6F D2A8AB     neq:    jnc     nsw         ;if d(i)<d(i+m) don't sw
                                                    ;change to jc for descending
                        ;
        AB72 C5         sw:     push    b           ;sw bytes not =
        AB73 46                 mov     b,m
        AB74 1A                 ldax    d
        AB75 77                 mov     m,a
        AB76 78                 mov     a,b
        AB77 12                 stax    d
        AB78 23                 inx     h
        AB79 13                 inx     d
        AB7A C1                 pop     b
        AB7B 0B                 dcx     b
        AB7C 78                 mov     a,b
        AB7D B1                 ora     c
        AB7E C272AB             jnz     sw
                        ;
                                ;str switched, ck if i1-m1<1
                        ;
        AB81 2A02AB             lhld    m1
        AB84 7C                 mov     a,h
        AB85 2F                 cma
        AB86 57                 mov     d,a
        AB87 7D                 mov     a,l
        AB88 2F                 cma
        AB89 5F                 mov     e,a
        AB8A 2A08AB             lhld    i1
        AB8D 19                 dad     d           ;if i1-m1<1 then jump to same
        AB8E D2A8AB             jnc     nsw         ;as no switch
                        ;
                                ;calc new d(i), d(i+m)
                        ;
        AB91 23                 inx     h           ;save new i1=i1-m1
        AB92 2208AB             shld    i1
        AB95 2A0EAB             lhld    di1         ;old d(i)=new d(i+m)
        AB98 EB                 xchg
        AB99 2A0AAB             lhld    ml1         ;addr offset
        AB9C 7B                 mov     a,e         ;new d(i)=old d(i)-offset
        AB9D 95                 sub     l
        AB9E 6F                 mov     l,a
        AB9F 7A                 mov     a,d
        ABA0 9C                 sbb     h
        ABA1 67                 mov     h,a
        ABA2 220EAB             shld    di1         ;save new d(i)
        ABA5 C35DAB             jmp     cp1         ;goto compare strings
                        ;
                                ;ck for j>k
                        ;
        ABA8 2A06AB     nsw:    lhld    j1
        ABAB 23                 inx     h           ;save new j=old j+1
        ABAC 2206AB             shld    j1
        ABAF 2208AB             shld    i1
        ABB2 EB                 xchg
        ABB3 2A04AB             lhld    k1
        ABB6 7D                 mov     a,l
        ABB7 93                 sub     e
        ABB8 7C                 mov     a,h
        ABB9 9A                 sbb     d
        ABBA DA18AB             jc      div         ;if j>k goto beginning divide m1
```

More

Listing 1 continued.

```
                             ;
                                        ;calc new d(j), d(i)
                             ;
      ABBD 2A0CAB                       lhld    dj1
      ABC0 D1                           pop     d
      ABC1 D5                           push    d
      ABC2 19                           dad     d       ;new d(j)=old d(j+1)
      ABC3 EB                           xchg
      ABC4 2A0AAB                       lhld    ml1     ;addr offset
      ABC7 EB                           xchg
      ABC8 C354AB                       jmp     lp2
```

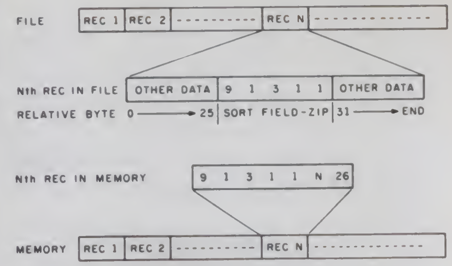


Fig. 3.

The program will then sort the records in ascending order and return.

●To sort in descending order, change the byte at location ORG+6FH (AB6FH) from D2 to DA before calling the program.

Test Results

To give times for comparison, I filled my memory with a random pattern of 0s and 1s. This minimum number of values for each location results in the maximum times for sorting. Test results are shown in Table 1a. The results for some of the same tests using the original algorithm are shown in Table 1b. The improvement is significant especially for large numbers since the new algorithm time is approximately a function of N log(N).

An Example

In general, a good technique to use is:

●For each record, load the field to be sorted into memory with a pointer. Fig. 3 shows a typical example.

●Sort using the routine given here.

●Use the now sorted pointers to access each record, in order.

As an example, suppose you want to do an alphabetical sort of names stored in a file in ASCII format (this is the normal string and file format of most languages). Getting the correct results is not difficult, but it does require some care. Consider the following correctly sorted names:

de la Fleur, Conrad
Diebold, Tom
DiGeorge, Olaf
DiGeorgeo, Dick
di Georgeo, Len
Di Georgeo, Pete
Dillingham, Sam

If you had sorted these names by just loading last name first name, into memory, you would have gotten:

Di Georgeo, Pete
DiGeorge, Olaf
DiGeorgeo, Dick
Diebold, Tom
Dillingham, Sam
de la Fleur, Conrad
di Georgeo, Len

To get around this problem, make the following changes as the names are loaded into memory for sorting:

●Convert all characters to lowercase (or uppercase).

●Remove all spaces and punctuation in the last name.

●Insert a delimiter between last name and first name with ASCII value lower than letters (I used a comma).

●Truncate this string plus the pointer to 16 bytes or pad with blanks if shorter.

For example, de la Fleur, Conrad, becomes "delafleur,conxx" and Jones, Sam, becomes "jones, sam00000xx," where xx is the pointer and the 0's are blanks. This allows 14 letters to sort on. With over 6000 names, I have not had a mistake caused by this length restriction; with less names you might get by with even less.

After you've done the above and sorted using this routine, you can use the now-sorted pointers to access the original records in alphabetical order.

Large Files

If the file is too large to be loaded into memory, even using the above method of loading only the sort field, then you need an additional technique. One method, if you don't know anything about the distribution of the field you're sorting on, is:

●Read into memory as above, stopping when you have filled memory.

●Sort this block 1 and save on disk.

●Read more fields in, sort and save as block 2, 3, etc.

●Merge the blocks by reading each block item by item and saving the appropriate entry on disk in a separate file. This works since each block is individually sorted.

This method works for files of any size, even ones that span multiple disks.

Conclusion

The sorting routine given here should let you sort your files on any field within a reasonable time. For those who wish, this program is available on a CP/M eight-inch disk or TRS-80 five-inch disk for $15 from Compleat Systems, 9551 Casaba Ave., Chatsworth, CA 91311 (213-993-1479).

The disk also contains a Z-80 version that takes 25 percent less space and runs 25 percent faster, and two additional utilities (search and move).■

AMOUNT OF MEMORY	# RECORDS	# BYTES/REC	TIME TO SORT SECONDS
16K	4K	4	22
16K	2K	8	22
16K	1K	16	15
32K	8K	4	65
32K	4K	8	75
32K	2K	16	54
32K	1K	32	34

Table 1a.

16K	4K	4	775
16K	2K	8	285
16K	1K	16	110

Table 1b.

MODEL II

26-4002
64K 1 Drive
$3440.00

26-4160 1 Drive EXP.	$1035.00
26-4161 2 Drive EXP.	1575.00
26-4162 3 Drive EXP.	2115.00
26-4501 Gen. Ledger	180.00
26-4502 Inventory	180.00
26-4503 Payroll	360.00
26-4554 Acct. Rec.	180.00
26-4701 Fortran	270.00
26-1157A Daisy Wheel	2290.00
26-1158 Daisy Wheel II	1799.00

$ DISCOUNT $

TRS-80® DEALER A301

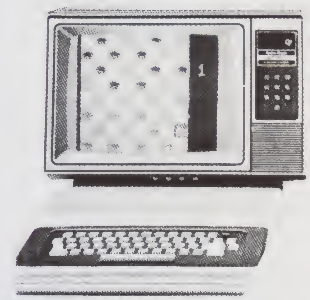
COMPUTER SPECIALISTS

26-1155 Quick Printer II	$187.00
26-1145 RS-232 Board	84.00
26-1140 "O" K Interface	249.00
26-1141 "16" K Interface	359.00
26-1142 "32" K Interface	469.00
26-1160 Mini Disk - Drive O	419.00
26-1161 Mini Disk - Additional	419.00
26-1154 Lineprinter II	699.00
26-1165 Line Printer V	1710.00
26-1159 Lineprinter IV	859.00
26-1166 Line Printer VI	1080.00
26-1563 Scripsit - Disk	79.00
26-1566 Visicalc	83.00
26-1562 Profile	72.00

MODEL III

26-1061 4K I	$630.00
26-1062 16K III	888.00
26-1063 32K III	
2-Drives, RS232	2225.00

COLOR



26-3001 4K	$360.00
26-3002 16K	540.00
26-3010 Color Video	360.00
26-1206 Recorder	54.00
26-3008 Joysticks	22.50

CENTRONICS

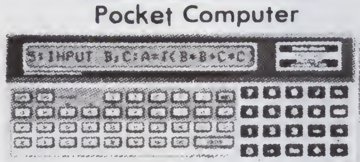
Fast 100 CPS Centronics 730 Printer	577.00
Text Quality Centronics 737 Printer	737.00

Model II Cobol Compiler
$360.00
Cobol Run Time Package
$36.00

ALL OTHER R.S. SOFTWARE FURNITURE, STANDS, CABLES AND ACCESSORIES AT DISCOUNT FROM CATALOG PRICE.

Novation Cat Modem	$149.00
CCA Data Management System	72.00
Adventure Games Games 1-9 each	14.00

Pocket Computer



26-3501 1.9K P.C.	$225.00
26-3503 Cassette I/F	45.00
14-812 Recorder	72.00

Acorn Software Products, Inc.

GAMES:	
Alien Invasion	$9.00
Stock Market	9.00
Star Trek	9.00
Block Em	9.00
Ting-Tong	9.00
UTILITIES:	
System Savers	14.00
EDUCATION:	
Language Teacher	18.00

FREE: COMPUTER CATALOG UPON REQUEST

1-800-841-0860 Toll Free Order Entry

MICRO MANAGEMENT SYSTEMS, INC. 100

No Taxes on Out Of State Shipments

Immediate Shipment From Stock on Most Items

DOWNTOWN PLAZA SHOPPING CENTER
115 C SECOND AVE. S.W.
CAIRO, GEORGIA 31728
(912) 377-7120 Ga. Phone No.

*TRS-80 is a registered trademark of the Tandy Corp.

R.S. 90 Day Limited Warranty
F-48 Form Provided

Largest Inventory In the S.E. U.S.A.

This super video game program runs on OSI systems.

Computerized Table Tennis, Anyone?

By Willis H. Cook

Everyone knows that a full-fledged computer can do a lot more than a video game device can, but can it do as much?

This program duplicates one of the original (if not *the* original) games introduced by the Atari Company several years ago. Judge for yourself how well this BASIC version performs. If nothing else, you will probably agree that imitating a fast-paced video game on a general-purpose computer is not a trivial task.

Program Features

The first requirement of the program was that rebounds had to be realistic. If the ball strikes a surface at, say, 20 degrees, it should rebound at 20 degrees. The program does this but it takes a lot of code; I'll discuss the procedure later.

The program will work for either one or two players. With two playing, your opponent scores a point whenever the ball reaches the border at your end of the screen. The scores are displayed just above the top of the border. The winner is the first one to reach a score of 15. My computer doesn't have joysticks, so I use designated keys on the keyboard to move the paddles up and down. Key 1 raises and key 2 lowers the left paddle, and keys M and N control the right paddle. Two two-handed players create a lot of activity on one keyboard, but the crowding adds to the fun.

When one person is playing, he uses the left-hand paddle. When the ball hits the border on the right it simply rebounds, as if against a backboard. The game ends after he has missed 15 balls. The computer then rates his play on a scale of 1 to 10.

The program also has a fast/slow feature. Implemented by means of a FOR. . .NEXT loop in line 450, the slow speed is slow enough for adults. Kids will naturally prefer the faster speed.

Serves are random, and originate from two different locations on the screen. Initially the ball was put into play from the center of the screen, but this meant that the player who

A typical game of computer pingpong. Contrary to what you might expect, there is a lot of body contact involved with this program.

Address correspondence to Willis H. Cook, 1298 Renee Drive, Lilburn, GA 30247.

A close-up of the screen display. Each paddle is individually movable. A game continues until one player reaches a score of 15.

took the serve did not have time to position his paddle correctly (unless he successfully guessed where the ball was heading). This condition seemed severe; with the current version of the program, if you have your paddle in the center of your court you can return any serve—if you're fast.

Details of Operation

The *OSI Graphics Manual* illustrates 12 directions of motion that can be implemented by means of FOR... NEXT loops using various step values (refer to Fig.1). For simplicity, I used only four of the 12 possible directions, represented by step values ±62 and ±66. If the ball is traveling along the +66 line and strikes a paddle, it should rebound along the +62 line. If, however, it strikes the bottom border, it should rebound on the −62 line. So, in order to determine the rebound, the program must know whether the ball has hit a horizontal or vertical surface. For rebounds from vertical planes, the sign of the step value remains unchanged.

There is also the special case (line 400) where the ball hits a corner: only the sign of the step value changes, not its magnitude. Lines 330 through 440 in the program determine rebounds. Line 290 checks the four adjacent locations surrounding the ball after

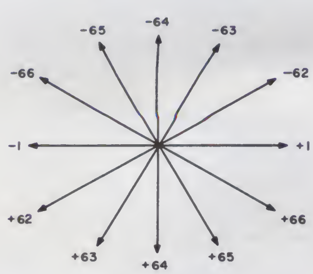


Fig. 1. Twelve directions of motion can be represented by changing the video display address by different increments.

Listing 1. Computer pingpong program in OSI BASIC.

```
10 REM  ********************
20 REM  *                  *
30 REM  *   PING  PONG     *
40 REM  *                  *
50 REM  ********************
60 REM
70 A=111:H=56900:F=57089:P=187:POKE F,0
80 G=161:M=54210:K=57088:Q=54236:POKE 530,1
90 U=0:V=0:U1=53508:V1=53529
100 FOR I=1 TO 28:PRINT:NEXT:POKE H,2
110 GOSUB 670: REM  INSTRUCTIONS
120 GOSUB 470: REM  SET UP BORDER
130 GOSUB 500: REM  SET UP PADDLES
140 REM  INITIAL TRAJECTORY
150 D=INT(4*RND(1)+1):ON D GOTO 160,170,180,190
160 B= 62:L=53711:GOTO 200
170 B=-62:L=54927:GOTO 200
180 B= 66:L=53711:GOTO 200
190 B=-66:L=54927
200 POKE L,A
210 REM  CHECK KEYBOARD FOR KEY CLOSURE
220 POKE K,128
230 IF PEEK(K)=128 THEN GOSUB 560
240 IF PEEK(K)= 64 THEN GOSUB 590
250 POKE K,4
260 IF PEEK(K)=  8 THEN GOSUB 620
270 IF PEEK(K)=  4 THEN GOSUB 650
280 W=PEEK(L+1):X=PEEK(L-1):Y=PEEK(L+64):Z=PEEK(L-64)
290 IF W=32 AND X=32 AND Y=32 AND Z=32 THEN 450
300 REM  'BLIP' SOUND
310 POKE F,50:FOR I=1 TO 10:NEXT I:POKE F,0
320 REM  CALCULATE REBOUND
330 C=ABS(B)
340 N=2*(C-64):C=C-N
350 IF X=P THEN 850
360 IF E=1 AND W=P THEN 410
370 IF W=P THEN 800
380 IF W=G OR X=G THEN J=J+1:GOTO 410
390 IF B<0 THEN B=C:GOTO 450
400 B=-C:GOTO 450
410 IF Y=P OR Z=P THEN B=-B:GOTO 450
420 IF B<0 THEN B=-C:GOTO 450
430 B=C:GOTO 450
440 REM  R DETERMINES THE SPEED.
450 FOR I=1 TO R:NEXT I
460 POKE L,32:POKE L+B,A:L=L+B:GOTO 220
470 FOR I=53568 TO 53598:POKE I,P:NEXT
480 FOR I=53632 TO 55050 STEP 64:POKE I,P:POKE I+30,P:NEXT
490 FOR I=55040 TO 55070:POKE I,P:NEXT:POKE 55104,32:RETURN
500 IF E=1 THEN 540
510 REM  RIGHT PADDLE
520 FOR I=54172 TO 54364 STEP 64:POKE I,G:NEXT
530 REM  LEFT PADDLE
540 FOR I=54146 TO 54338 STEP 64:POKE I,G:NEXT:RETURN
550 REM LEFT PADDLE UP
560 IF M=53698 THEN RETURN
570 M=M-64:POKE M-64,G:POKE M+192,32:RETURN
580 REM  LEFT PADDLE DOWN
590 IF M=54850 THEN RETURN
600 M=M+64:POKE M+128,G:POKE M-128,32:RETURN
610 REM RIGHT PADDLE UP
620 IF Q=53724 THEN RETURN
630 Q=Q-64:POKE Q-64,G:POKE Q+192,32:RETURN
640 REM  RIGHT PADDLE DOWN
650 IF Q=54876 THEN RETURN
660 Q=Q+64:POKE Q+128,G:POKE Q-128,32:RETURN
670 PRINT TAB(8)"WELCOME TO":PRINT:PRINT
680 PRINT TAB(5)"P I N G   P O N G":PRINT:PRINT
690 PRINT:INPUT"HOW MANY PLAYERS ARE THERE";E
700 C$="":IF E=2 THEN C$=" LEFT"
710 PRINT:PRINT"1 MOVES THE";C$;" PADDLE UP."
720 PRINT"2 MOVES IT DOWN.":PRINT:IF E=1 THEN 750
730 PRINT:PRINT"N MOVES THE RIGHT PADDLE UP."
740 PRINT"M MOVES IT DOWN.":PRINT
750 R=20:PRINT"DO YOU WANT A FAST"
760 INPUT"OR A SLOW GAME";C$
770 IF LEFT$(C$,1)="F" THEN R=0
780 FOR I=1 TO 1500:NEXT I
790 FOR I=1 TO 28:PRINT:NEXT:RETURN
800 U=U+1:POKE U1,INT(U/10)+48
810 IF U<10 THEN POKE U1+1,U+48:GOTO 830
```

More

Listing 1 continued.

```
820 POKE U1+1,U+38
830 POKE L,32:IF U=15 THEN 940
840 GOTO 890
850 V=V+1:POKE V1,INT(V/10)+48
860 IF V<10 THEN POKE V1+1,V+48:GOTO 880
870 POKE V1+1,V+38
880 POKE L,32:IF V=15 THEN 940
890 FOR I=1 TO 50:POKE K,128:IF PEEK(K)=128 THEN GOSUB 560
900 IF PEEK(K)=64 THEN GOSUB 590
910 POKE K,4:IF PEEK(K)=8 THEN GOSUB 620
920 IF PEEK(K)=4 THEN GOSUB 650
930 NEXT I:GOTO 150
940 IF E=1 THEN 960
950 IF U>V THEN T1=54020:GOTO 970
960 T1=54035
970 IF E<2 THEN 1010
980 FOR I=1 TO 7:READ T:POKE T1+I,T:NEXT
990 FOR I=1 TO 2000:NEXT I:POKE 530,0
1000 FOR I=1 TO 28:PRINT:NEXT:POKE H,1:END
1010 FOR I=1 TO 7:READ T:NEXT:FOR I=8 TO 18:READ T
1020 POKE T1+I-8,T:NEXT:J=INT(J/15)
1030 FOR I=1 TO 2000:NEXT I:FOR I=1 TO 28:PRINT:NEXT
1040 PRINT"ON A SCALE OF 1 TO 10"
1050 PRINT"YOU RATE A";J:GOTO 990
1060 DATA 87,73,78,78,69,82,33,69,78
1070 DATA 68,32,79,70,32,71,65,77,69
```

```
CHANGE LINE 80 TO

80 G=161:M=54210:K=57088:Q=54236:POKE 2073,96


CHANGE LINE 990 TO

990 FOR I=1 TO 2000:NEXT I:POKE 2073,173
```

Listing 2. Line changes for disk systems using OS-65D, V 3.2.

A	ASCII code for a lowercase "O." Used for the ball.
H	Output display format, 32×32 or 32×64.
F	Sound port.
P	Video display character code for a gray display cell. Used for border.
G	Video display code for a solid white display cell. Used for paddles.
M	Initial position of left paddle.
K	Keyboard port.
Q	Initial position of right paddle.
U	Left-side player's score.
V	Right-side player's score.
U1	POKE location for left player's score.
V1	POKE location for right player's score.
I	Miscellaneous counter.
D	Random integer between 1 and 4. Determines initial ball location and trajectory.
B	Display address increment. Determines the ball trajectory.
L	Instantaneous ball location.
W	Location just to the right of the ball, i.e., L+1.
X	Location just to the left of the ball, i.e., L−1.
Y	Location just below the ball, i.e., L+64.
Z	Location just above the ball, i.e., L−64.
C,N	Temporary variables used in calculating B after a rebound.
R	Speed controller.
E	Number of players.
T1	POKE location for end-of-game message.
T	ASCII code data for end-of-game message.

Table of variables.

each move to see if the ball has hit anything. If it has, control passes to the rebound section.

If the procedure sounds complicated, bear in mind that during the actual programming I used about two percent analytical reasoning and 98 percent trial and error. While trying to perfect the rebound algorithm, I was working with only those parts of the program that set up the border with the bouncing ball inside. Every now and then the ball would break through the border and bounce around between the border and the edge of the screen. And sometimes it bounced completely off the screen and into the program somewhere. (Instead of poking to a video display address, I was poking to the RAM where the program was stored.)

Also notice lines 510–660, which control the paddles' movements. You have to stop moving the paddle when it gets to the top or bottom of the screen or you might push it right into the program.

Line 310 causes a "blip" sound whenever the ball hits the border or a paddle. This uses the computer's programmable audio output with an external audio amplifier. Early models of the C2-4P do not have sound output, but the program can be run unchanged on these machines.

Lines 470–490 create the border. If you decide to change the size of the border, be sure to change the dimensions in multiples of two or you will experience the breakthrough problem mentioned earlier.

Lines 220–270 scan the keyboard repeatedly to see if either player wants to move his paddle. Between turns, also, the players must be able to move the paddles to anticipate the next serve. So, rather than waste the interval in a do-nothing loop, lines 890–930 scan the keyboard again.

When using the keyboard-scan feature, the control-C (interrupt) function must be disabled. Line 80 disables control-C, and line 990 restores the function when the program ends. This is done differently on BASIC-in-ROM and disk systems. Listing 1 is for ROM-based machines. If you use disk BASIC with the OS-65D version 3.2 operating system, change the lines as shown in Listing 2. If you use a different disk operating system, check your manual for the proper control-C enable/disable techniques.

Evaluation

After running this program you

will probably agree that it has two shortcomings compared to the original. One is execution speed. Although the combination of 6502 CPU and Microsoft BASIC is one of the fastest around, things slow down a bit when two players must maneuver their paddles simultaneously.

The program must scan the keyboard, move the paddles and the ball, and then check at least six display memory addresses before moving the ball again. I don't think it is an objectionable problem but it certainly is noticeable. A possible solution, short of rewriting the entire program in assembly language, would be to translate the paddle-move routines into assembly language and access them with the USR function.

The other problem is a lack of resolution in the 32 × 32 display format. Because the ball always travels on a low-angle trajectory, it moves two display cells horizontally for each cell it moves vertically. Each cell is relatively large, so the apparent motion is rather jumpy. I originally wrote the program for the 32 × 64 format and the resolution was better, but there were a couple of other drawbacks. I never could get the border centered on the screen, and the pingpong ball trajectory tended to repeat itself with predictable uniformity. Thus, the 32 × 32 format was a compromise.

Nevertheless, the end result is reasonably successful. Hopefully, some of the techniques used in this program will turn up again in other applications. And the exercise has at least given me a healthy respect for video display programming. ■

OSI Clear-Screen Command

By Cyril Bradshaw

```
RUN"CHANGE","PASS"

DISK CHANGE UTILITY

MODE: HEX(H), DEC(D) ? H
UNIT ? A
ADDRESS OFFSET ? C00
ADDRESS ? 2DCE
00002DCE    00 ? A2
00002DCF    00 ? D8
00002DD0    00 ? A9
00002DD1    00 ? D0
00002DD2    00 ? 85
00002DD3    00 ? FF
00002DD4    00 ? A9
00002DD5    00 ? 00
00002DD6    00 ? 85
00002DD7    00 ? FE
00002DD8    00 ? A8
00002DD9    00 ? A9
00002DDA    00 ? 20
00002DDB    00 ? 91
00002DDC    00 ? FE
00002DDD    00 ? C8
00002DDE    00 ? D0
00002DDF    00 ? FB
00002DE0    00 ? E6
00002DE1    00 ? FF
00002DE2    00 ? E4
00002DE3    00 ? FF
00002DE4    00 ? D0
00002DE5    00 ? F5
00002DE6    00 ? 60
00002DE7    00 ? .
ADDRESS ? 220E
0000220E    A5 ? CD
0000220F    09 ? 2D
00002210    A5 ? .
ADDRESS ? 231A
0000231A L  4C ? 43
0000231B E  45 ? 4C
0000231C    D4 ? D3
0000231D G  47 ? X
```

Listing 1. OSI disk modification.

My C2 8P had everything, except a decent screen-clear command. The FOR: NEXT loop took forever to execute. (Actually, it takes one second, but it seems like forever.) It scrolled the copy off the screen and wasted time that I could have been using to enter new data, and caused me much mental anguish. Lesser breeds of computers could blank the screen with a simple command—it was downright embarrassing.

(I understand why Ohio Scientific didn't include the CLS command in OS-65U. The C3 with a serial terminal doesn't need it, since the terminal has, self-contained, all the operating language it needs. This is just for the C3's poor relatives.)

A peek into the OS-65U language (system) showed some unoccupied space on the disk where a machine-language routine could be tucked away. But how would I command the routine? What could I use for a command word, and where could it be stored? A solution to this was to find the least-used reserved word and substitute my choice for it.

That was easy. The least-used reserved word—actually, I never use it —is LET. Its use is optional, even superfluous, since LET A=B and A=B are equivalent statements. Why not replace the reserved word LET with CLS, a commonly used command for clearing the screen?

Modification

Some of the empty language (system) areas on the disk are not suitable. They won't accept the routine and permit it to function properly due to program interferences or other problems peculiar to the language. By trial and error, I found that $2DCD and on was a good residency for the program. The easiest way to enter the routine is through the OS-65U Disk Change Utility program.

To enter the routine and modification to your disk, call in the Change program by entering RUN "CHANGE","PASS." Use hexadecimal addresses; i.e., enter H in response to MODE: HEX(H), DEC(D)?. Then select the disk drive containing the disk whose language you are about to change; i.e., UNIT ? A (or B as the case may be). In response to ADDRESS OFFSET ? enter C00, and to the inquiry ADDRESS ? respond with 2DCE.

Follow the procedure in Listing 1. Carefully enter the values in the column to the right of the ? as each address appears (being certain of the sequence) and enter a period (.) at the indicated address and the new address as called for. An X exits from the program, followed by CLOSE. This will enter CLS for LET and redirect it to the routine you have just entered.

Reboot the system into the machine, and CLS will then give the results you desired.

Your language now contains CLS, and if any of your programs had LET in them, it will now be CLS A=B, or

Cyril Bradshaw, PO Box 188, Colorado City, AZ 86021.

some permutation of this. Just find the CLS commands (the old LET commands), or let your computer find them for you, so that they can be changed. To have your computer find them, run your program, and when you come to that former LET command, the screen will clear and give you a ?SN ERROR and the offending line number.

Now you need to list your programs, find the offensive FOR C=1 TO 30 : PRINT : NEXT loops in them and replace them with CLS. Now your screen will clear in .022 second, instead of one second with the old FOR:NEXT loop scrolling screen clear time.■

Call For Manuscripts

Kilobaud Microcomputing is looking for business articles!

Businessmen in all fields are beginning to take notice of the microcomputer. They are eager to know which computers, peripheral equipment and applications software will let them take full advantage of this new tool. What knowledge do you have to share?

Here are the kinds of articles that we want you to write for us:

- Are you a businessman with a system up and running? We want to know how it works. What were your expectations? Have they been fulfilled? Did you find the software that you wanted? What problems have you had? How did you overcome them? What recommendations do you have for other businessmen?
- We want reviews from a businessman's perspective of specific hardware and software. If you've recently bought a new product and want to tell others how great—or poor—it is, *Microcomputing* will provide you with a forum.
- What programs have you written to meet your specific needs? Perhaps another businessman can use them, too. Even if he can't, your program may serve as a springboard for other ideas.
- Perhaps you aren't using your micro for business, but know a company that is. Trot on down with your pencil and notebook, and find out what they're up to. While they might not have the time to write up their experiences, they might be more than willing to tell somebody else about them. And an outside observer will often be able to see things with a unique and valuable perspective.

Don't worry if you're not a professional writer. That's what we editors are here for. And we'll be more than happy to send you a copy of our writer's guidelines.

Send your manuscripts and correspondence to:
Kilobaud Microcomputing
Pine St.
Peterborough, NH 03458

Computer Stations and the Paper Tiger co-star in this gripping drama about hard-copy Apple graphics.

A Tale of Two Screen Dumps

By Jim Hansen

One of the remarkable features of the Apple computer is its high-resolution graphics. Indeed, there are many accessory devices on the market today that exploit this capability. These range from television interfaces for digitizing and displaying a television image on the Apple screen to digitizing pallets that can be used to make digital copies of maps, plans, pictures and the like. Graphics images can also be generated under program control within the Apple itself.

Images can be put into the display memory and stored on disk. But sometimes you'll want a hard copy. You can get it by either taking a photograph of the screen or dumping to a printer with graphics capability. Strangely, Apple has ignored the problem of getting hard copies, and Apple users are left to resolve the situation on their own.

Operation of Paper Tiger graphics (the original Computer Stations screen dump program). Two versions of the screen dump are provided on the disk of the original screen dump. One was written to run with Applesoft, the other was for integer BASIC. Operation for both versions is the same. When run, high-resolution screen number one is displayed, and a series of questions answered with "Y/N" are presented. Loading the program wipes the high-resolution screen, leaving the image shown above.

Vendors of screen dump programs seem to support the Integral Data Systems 440G Paper Tiger Printer the best. Computer Stations, Inc. (12 Crossroads Plaza, Granite City, IL 62040), sells two such programs. Both are available at many computer stores or from Computer Stations.

The programs run on either the Apple II or Apple II Plus. The original, Paper Tiger Graphics Software, has been available since 1979. Enhanced Paper Tiger Software was released last year.

To use either package, your Apple must have at least one disk drive. No specific memory requirements are spelled out in the manuals supplied by Computer Stations.

Two Complete Programs

Both packages are complete programs in themselves. When placed in the master disk drive and booted, they ask for information regarding the type and location (slot number) of your printer interface. The enhanced version lets you use six different popular interface cards. The original package works only with the Apple Centronics parallel card or the Apple high-speed serial interface.

The older version operates in the "20 question" mode; that is, it solicits information by a series of questions answered yes or no. It can operate only on the first high-resolution page of the Apple and can print either a

Jim Hansen, Methodist Hill, Lebanon, NH 03766.

positive or negative image of the graphics display. This is necessary since it is unlikely that you will want a graph printed black with white dots; photographic pictures, on the other hand, must be printed this way or you will get a negative image. The program can also double the size of the printed image. Generally, it will prompt you for a file name, ask how you want the picture printed, print it and recycle.

The enhanced package presents two menus. The options available from the first menu at the beginning of the program run select the type of interface and set slot number. The second menu then appears and remains until you are finished. It lets you dump either high-resolution screen; choose regular or double size printout; position the picture on the left, right or center of the printed page; choose reversed or normal image (plot/picture mode); display the selected high-resolution memory on the screen; or display the disk catalog.

Option selection is simple. Each item on the menu is numbered. The selected mode for each item is displayed on the screen using reverse print. To switch to another mode you type the number of the item. The selection automatically moves to the next available option after each menu item. It is a simple and convenient way to select any combination of about a dozen options or commands.

The enhanced package offers several benefits over that of the less expensive, older version. It lets you use more (and in some cases less expensive) interfaces. It can dump either high-resolution page instead of only the first one, and it can position the picture on the printed sheet. Finally, it is easier and more convenient to use, since it uses a menu and removes most of the vestiges of Teletype data entry methods.

Both programs work without error. Interestingly, the folks at Computer Stations found that by setting the Paper Tiger to the 50 Hz power selection and using the double-sized printout, the horizontal and vertical dot resolution becomes more closely matched (they claim .7 percent). Thus, a circle on the Apple screen will print out as a circle instead of the usual goose egg. This will only work if you *set the 440 to* 50 Hz and run it at 60 Hz, and thus will probably apply only to North American countries. Those in Europe and elsewhere served with 50 Hz power will not be able to use this "feature." (I have not tried this out and so can't verify its operation.)

Personal Comments

I have used the older version for some time now and have no complaints. Generally, images must be saved to disk before they can be printed, since loading a program from DOS wipes out the high-resolution screen. If you have not seen the enhanced version run, chances are you will be satisfied.

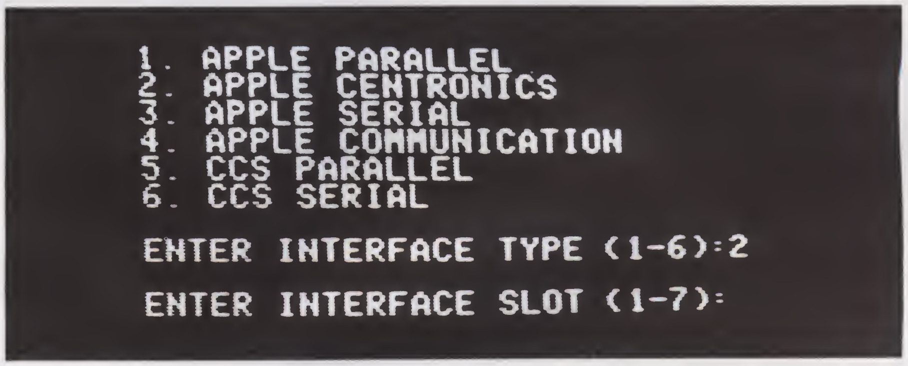
The biggest improvement in the enhanced version is that you can choose from a wider variety of interface cards. The older version, as mentioned before, lets you use only two of Apple's cards. The only other feature that influences the actual operation is that the picture can be printed on the left, center or right side of the page. This is nice to have, but I got the same results by moving the paper tractors.

Another impressive and convenient feature is the menu-driven option selection. I was impressed with the various human engineering features of this software package. Simple concise directions—either of the menus or the catalog—are presented as required with every display; all operations are easily controlled; and the information is not ambiguous, extraneous or misleading.

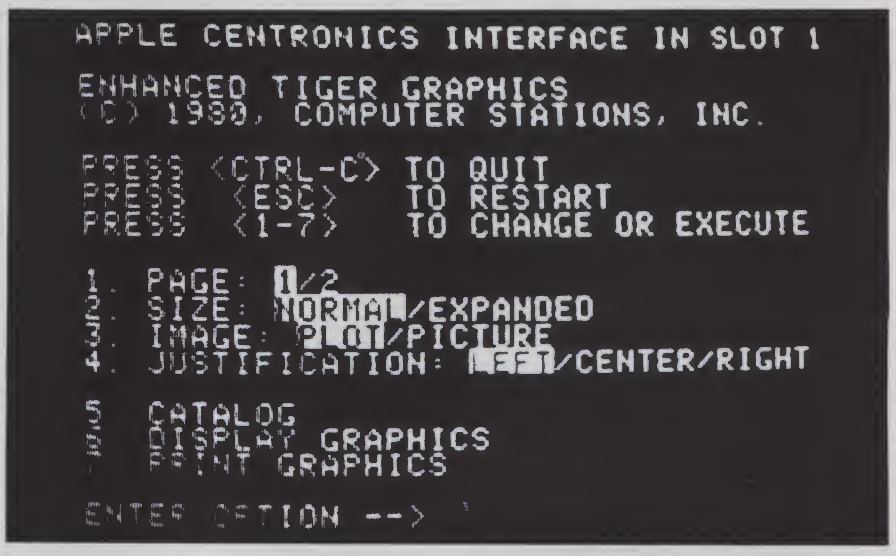
What You Get for Your Money

The enhanced version, which comes on a disk, includes several demonstration pictures and the instruction manual, as brief as the old

If a catalog is requested, a program from it may be loaded after the prompt for "Picture Name." The program then asks whether the image is to be inverted (printed positive or negative) and whether a double size (EXPAND SCREEN) print is to be made. Printout takes place after the expand screen question is answered.



Operation of the Enhanced Paper Tiger Graphics Software. When run, a menu listing the available interface drivers is displayed and the user is asked to type in the number corresponding to the type of interface in use. Next, the slot number of the interface is requested.



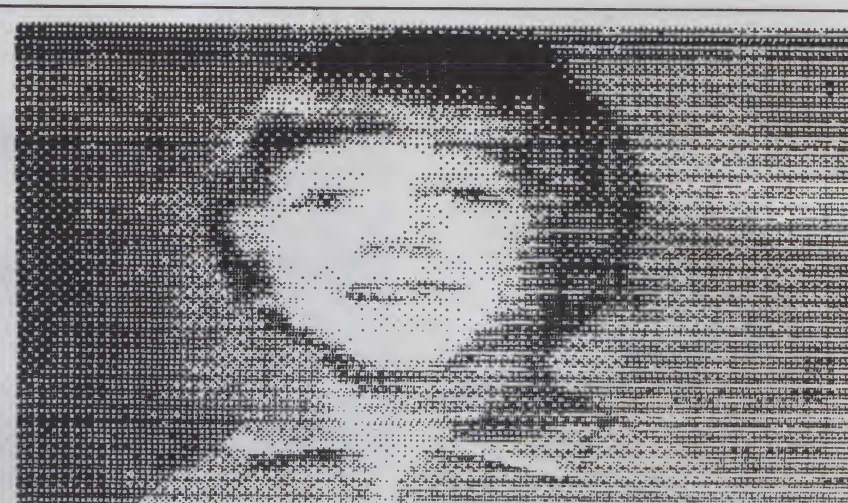
After the slot number is entered, a second menu is displayed. Any of the options may be changed or commands issued by simply typing the number of the menu item. Notice that complete operator instructions are displayed along with the menu. If a catalog is requested, the menu disappears and is replaced with the catalog. The desired program from the catalog is selected by positioning the cursor next to the file name using the space bar, then typing a return. The file is then loaded into screen memory and the catalog is replaced by the menu. If command 6 (DISPLAY GRAPHICS) were issued by typing a 6, the high-resolution page selected by item 1 (PAGE) will be displayed until any key is typed, at which time the menu is restored. The image is printed when command 7 is typed.

manual, but also describes how to save a picture to disk and gives connector wiring information for several interfaces.

The old software package includes two versions of the screen dump program. One is for use with Applesoft, the other with integer BASIC. The manual, though adequate, is not as elaborate or detailed as that sold with the enhanced screen dump.

Conclusions

Is either worth the price? If you want a commercial screen dump program for the Apple that works, and works well, either one will do. The original price for the old screen dump was $24.95; it is now valued at $34.95. The enhanced version is $44.95. The Scotch in me says that these prices are a bit steep for what would involve a couple of weeks' work after supper if you were to write the same programs from scratch. But you get what you pay for; in this case, fully debugged programs, each of which does what it is advertised to do with a frustration factor of approximately zero. ■



Printed picture done on an IDS 440G printer.

✓Reader Service—see page 210

GET ORGANIZED

With These Utility Programs From Instant Software

FIND IT QUICK

Put an end to the Misplaced Information Syndrome! Here is a reliable, fast, subject-oriented information locator—FIND IT QUICK! This information indexing and retrieval system is versatile and flexible—the ideal tool for doctors, lawyers, engineers, businessmen, educators, anyone who needs to locate information.

Designed with a journal/magazine format, FIQ is versatile enough to be used to store and retrieve any type of reference information including client lists, vendor advertisements, report bibliographies and more—even poems and famous quotations. Subjects can be indexed or called according to author, journal and date or by up to nine keywords that you determine. Keywords can be used singly or in combination during the search routines.

FIQ can be used by anyone. It requires no special skill, just the ability to type. Article references to be stored can be preselected and then entered into the system by a secretary or assistant. The contents of entire file cabinets can even be indexed and stored!

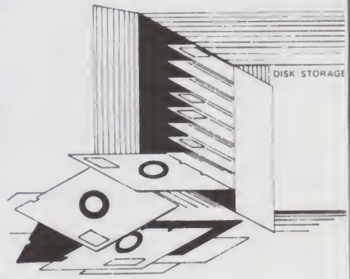
FIND IT QUICK requires a minimum system consisting of a TRS-80 Model I Level II with 16K of RAM, an Expansion Interface with 16K of RAM and at least one disk drive. Up to 1080 items can be stored on one data disk. Instant Software's Tiny DOS operating system is included on the program disk so that you can use the program without fuss or bother.

Make your computer work like your own personal librarian with FIND IT QUICK. Can you afford to wait another day?

Order No. 0258RD $49.95. Disk

MASTER DIRECTORY

The MASTER DIRECTORY is a disk file storage program that reads the files on all your disks and stores the file names and extensions and even records the free space on each disk. All you have to do is number the disks in your library and the MASTER DIRECTORY will keep track of their contents. You can read the names, displayed alphabetically, search the DIRECTORY for file names and extensions, delete disks and search for free space. You can store 5000 files or 320 disks, whichever comes first.



Your disk storage problems are over now that the MASTER DIRECTORY is here. This package requires the following minimum system:

1. A TRS-80 Level II with 16K RAM.
2. An Expansion Interface.
3. One (or more) disk drives.

Order No. 5005RD (disk-based version) $29.95

40

Instant Software™

PETERBOROUGH, N.H. 03458

TO ORDER:

SEE YOUR LOCAL INSTANT SOFTWARE DEALER OR CALL TOLL-FREE

1-800-258-5473

LOWEST PRICES ON THE TRS-80

Computers

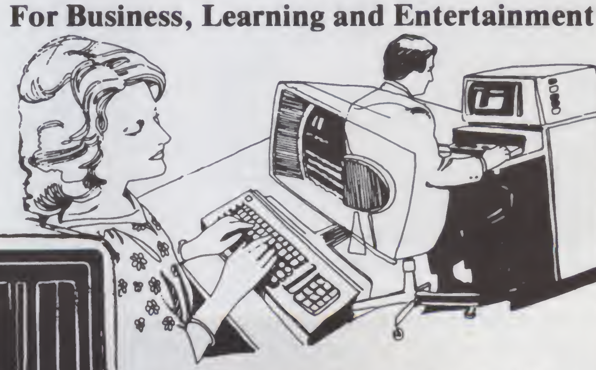
For Business, Learning and Entertainment

Pan American Electronics

71

INCORPORATED

a Radio Shack AUTHORIZED SALES CENTER



For More Information Call

TOLL FREE ORDER NUMBER
800-531-7466
Texas & Principal NUMBER
512-581-2765
Telex Number 767339
Dept K, 1117 Conway
Mission, Texas 78572

Even inveterate hardware tinkerers can appreciate the capabilities of Wordstar on Exidy's Sorcerer.

Word Wizardry

By Charles H. Strom

The Exidy Sorcerer computer is a single-board machine based on the Z-80 microprocessing chip. It is similar to the TRS-80 or Apple in appearance; it comes as a single unit with an integral keyboard, and the user must supply a video monitor and tape recorders.

The Sorcerer is equipped with a dual cassette interface capable of transferring data at 300 or 1200 bps (bits per second). It also has a serial RS-232 interface, actually identical to the cassette port, and a parallel, Centronics-compatible, interface.

As an inveterate hardware hacker, I of course looked for ways to expand the system. I had considerable experience with disk-based machines and the CP/M disk operating system, the de facto industry standard. I don't need to tell anyone the advantages of a disk system.

Exidy was wise in designing S-100 compatibility into the Sorcerer. It was my reason for buying the unit. The expansion box contains a power supply that supports the six card-edge connectors and all circuitry necessary to allow translation between the Sorcerer bus and the proposed IEEE-defined S-100 bus. Refer to the excellent Exidy technical manuals for a description of the bus differences. Exidy does supply two disk systems that plug in directly to the expansion interface edge connector on the Sorcerer, but I opted for eight-inch disks for several reasons; I wanted the S-100 flexibility, and the cost of an eight-inch system configured by myself from several sources was no more expensive than buying a ready-to-run five-inch system from Exidy.

My hardware consists of a Micromation disk controller and a single Shugart 801 drive, power supply and enclosure from Morrow/Thinker Toys. The controller has an on-board PROM specifically designed for the Sorcerer. It resides at the ROM-Pac address space; thus, when the system is powered up, the disk automatically boots up the operating system.

The controller allows a mix of single- or double-density drives, and the density is automatically determined when a disk is booted. Micromation is a company to respect—both their hardware and software are exceptional.

The CP/M disk operating system has a hardware-dependent portion called BIOS (basic input-output system), which is the only part that is changed when adapting CP/M to different hardware configurations. Some vendors supply a skeleton BIOS that allows little flexibility without reprogramming, but Micromation supplies a fully-commented, well-implemented source listing. Morrow, from whom I obtained the drive itself, is one of the oldest and most respected names in hardware design for micros, with low prices. What more can you ask? I also bought two 8K Econoram-II static memory boards from Godbout (Computhink), another pioneer in excellent design of hardware. The Sorcerer uses dynamic memory in the machine itself, but to eliminate any possible problems with memory in the expansion box, I selected a static design.

To run my word processing package, I have a DTC-300 daisy-wheel terminal interfaced to the serial port supplied on the Micromation disk controller. This is a fully-implemented serial port with baud rate selection and handshaking. The Micromation BIOS uses this serial port to run the CP/M list device, so those of you who are neophytes at assembly-level programming need not fear.

I might mention that CP/M, although very versatile, is not simple to use for a beginner. The documentation leaves something to be desired, and reminds me of IBM manuals. Modification is even more difficult and is not recommended for a newcomer to assembly language. Once you gain some experience with it, however, it shows itself to be a fine product.

As I previously noted, CP/M has become a sort of standard; the vast majority of disk-based software available for 8080- or Z-80-based machines running with an S-100 bus is designed to run under CP/M. Other operating systems—none of which I have used—may be excellent products, but they are of questionable use to a user who wants to run a variety of software designed by a third party.

Wordstar

Wordstar, a product of Micropro, was designed to run under CP/M. The version I use is 93A, which re-

Address correspondence to Charles H. Strom, 100 Bleecker St., #4C, New York, NY 10012.

quires 44K of memory. The explanation for this rather large memory requirement will become clear as I describe the wealth of features available under Wordstar. However, before you can edit a document, the program must be installed, or patched, for the specific hardware configuration. Micropro supplies a program, called Install, which automatically does just that for the most common terminals and printers, but the Sorcerer computer and the DTC-300 printer are not included.

Basically, I must configure the program for the Sorcerer as a memory-mapped video device. A series of patches are required for screen width, number of lines, starting address of the video RAM space and so on. This is straightforward and clearly documented. Wordstar employs reverse video for menu display, marking sections of text prior to moves, etc., and this creates a problem.

The Sorcerer does not have a reverse video (black-on-white) character set. The solution is to insert a jump at the Wordstar terminal initialization location (TERMINI) and to write a simple machine-language routine to generate the inverse video characters. This is simpler than it sounds; the standard ASCII character set is defined starting at location F800 hex. Each location is complemented and the result is stored in the user-defined graphics space (starting at FC00 hex) such that by setting the high-order bit of the character, the inverse character is addressed.

This procedure need only be performed when initially loading the program, since the clear screen command, which resets the Sorcerer's graphics, is not generated when Wordstar is used in the memory-mapped video mode. According to Micropro, Wordstar also requires cursor move routines, but I see no difference with or without them.

Wordstar is designed for installation of the more popular printers, such as any Teletype-like device, a selectric-like printer or the various daisy-wheel printers. The latter allow more complicated functions such as microspace justification and super and subscripting, but boldface and underlining can be supported with dumb printers as well. My printer is an oldie, and in fact uses the Intel 4004, the first microcomputer chip, as a controller. It was a simple matter to patch some of the daisy-wheel features, but I have no way of locating the area in Wordstar responsible for microspace justification. I just ordered the customization notes, and at $100, they had better supply the needed information!

Before going on with a general description of Wordstar, I need to discuss a few problems associated with this hardware-software mix. The designers at Exidy appear to have a philosophy that encourages the use of firmware (software burned into a read-only memory device) rather than hardware. The serial port does not have its own UART, and more importantly, the keyboard is not hardware-encoded. This means that every time a key is pressed, the CPU must scan the keyboard and then generate the proper character. This is a short-sighted design, for the lack of a two-dollar encoder chip makes the keyboard excessively slow. Wordstar polls the keyboard a certain number of times as a timing loop when menus are displayed, the cursor is blinked, etc.; as a result, there is a long wait when this loop is executed. I have been unable to get a fix from Micropro.

A more serious problem becomes apparent when the user reaches the end of a text line. The program has default settings of word-wrap, which means if a word is longer than the margins allow, it will automatically be moved to the next line, and right justification, which means spaces are inserted in the line so that the right margin is even throughout the document. (When printing, if microspace justification is implemented, the spacing is distributed evenly for a better-looking document.)

The Sorcerer keyboard has no rollover; in other words, when the previous line of text is being automatically rearranged, characters are lost if the typist is inputting keystrokes at even a rather low rate. This is not the fault of Wordstar, which, when used with a normal keyboard, operates without losing any characters. I've found two fixes.

The word wrap and justification features may be toggled off, and text may be entered as a continuous string. It is a simple matter to turn these features on again at the end of the session and reform the document with the ↑B command. (↑ means pressing the control key simultaneously with the other key.) The other solution is to implement the DTC-300 daisy-wheel terminal keyboard as the

console input device. This keyboard has a standard hardware encoder, and Wordstar thus operates without any of the aforementioned problems.

Other Features

A menu, displayed at all times, reminds the user of the most-used commands such as cursor move, character, word and line deletes, etc. Also, several secondary menus can be called up to explain other commands, and, in the case of two-letter commands, automatically are called when the user pauses before entering the second command letter. It is obvious that a great deal of thinking went into the design. The menus can be suppressed if you want to use the entire screen.

The manual, incidentally, is very complete and was written with the Wordstar program. It is an excellent example of what Wordstar is able to do, and also shows Micropro's faith in their product. There is a complete index of commands as well.

Commands such as "find" or "replace a character string" are practically self-explanatory, but the manual comes in handy to detail the various options built into these commands. Once again, the explanations can be called up on the screen. Incidentally, the messages occupy a separate file called WSMSGS, which is a respectable 19K size. Another valuable feature allows the input of another disk file at any desired place with a single command. Thus, boiler plate, or preformed, text can be inserted in another document.

Rather than go on about the many dozens of additonal features, I recommend that you buy the manual. This should be a standard procedure when you plan to spend almost $500 for a software product.

One feature curiously absent from Wordstar is a simple way of preparing a series of letters from a mailing list. Micropro suggests putting a special symbol such as $ in place of an inside address, for example, and then calling in the real address from an external file and using a replace command to place the address at the specified location. This sounds like an inconvenient, though workable, solution. There is an enhanced Wordstar now available that has more convenient mailing list features in it, but I don't feel like spending more than $100 extra just for this feature. There are custom mail list programs available for just such a purpose.

Conclusion

I'm quite pleased with Wordstar. I've been exposed to most of the dedicated word processing systems available in the commercial market, and although the use of dedicated keys rather than control keys makes them simpler to master, I don't really feel that they have much of an advantage as the user becomes more practiced; the alternative of a package such as Wordstar allows purchase of much more general-purpose hardware that can be adapted to tasks other than word processing. The Sorcerer has also proved of great utility; I only wish that Exidy had included more hardware in the keyboard. Well, if things turned out perfectly, I would be miserable. After all, what good is a microcomputer system if you can't spend enjoyable hours trying to wring out better performance from it?■

AUTOMATED EQUIPMENT INCORPORATED

18430 Ward, Fountain Valley, CA 92708

Call Toll-Free for latest low prices!
1-800-854-7635 Outside CA
714-963-1414 Inside CA
1-800-854-7635 Outside Cont. USA
Prices change daily to meet competition.

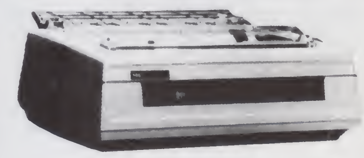
TERMINALS

TELEVIDEO 912 B	$665
TELEVIDEO 912 C	665
TELEVIDEO 920 B	945
TELEVIDEO 920 C	715
SOROC IQ-120	675
ZENITH A-19	735
HAZELTINE 1500	850
TELEVIDEO 950	CALL

NORTHSTAR

HRZ-1D-32K	$1975
HRZ-2D-32K	2270
HRZ-1Q-32K	2240
HRZ-2Q-32K	2650
ADDITIONAL 16K RAM	365
ADDITIONAL 32K RAM	475
HARD DISK SYSTEM	3900

SPECIAL MEMORY CONFIGURATIONS AVAILABLE, INCLUDING SYSTEMS GROUP.



PRINTERS

NEC 5510 (TRACT., RIB., THIM)	$2540
NEC 5515 (TRACT., RIB., THIM)	2600
NEC 5520 (TRACT., RIB., THIM)	2900
NEC 5530 (TRACT, RIB., THIM)	2540
MALIBU	1800
TI-810	1495
TI-820	1625
EPSON MX 80	CALL
EPSON MX 70	CALL
ANADEX DP-9500/9501	CALL

VIP's Call A.E.I.

VIP's customarily need top quality and service within stringent budgetary limits. naturally, they call A.E.I.

Item: A.E.I. tests virtually every* item sold, before shipping.

Item: A.E.I. initializes every piece of software, and makes a copy. If you have a problem, A.E.I. can check to locate the problem by using the in-house duplicate, saving you time.

Item: A.E.I. has in-house expetise on every item offered.

Item: A.E.I. prices are competitive with the lowest, no-service, no in-house expertise dealer.

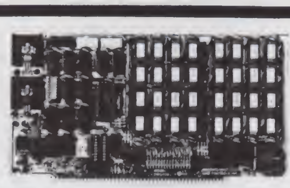
40% OF ALL A.E.I. SALES ARE TO PUBLIC AND SEMI-PUBLIC INSTITUTIONS.

A partial list** of A.E.I. customers who purchased during the last quarter of 1980 includes: University of Nebraska, University of Virginia, U.S. Dept. of Interior, University of Kentucky, University of California, Massachusetts Institute of Technology, U.S. Air Force, Naval Air Development Center, Brown University, University of Oregon, University of Utah, Ohio State University, University of Southern California, California State Dept. of Water Resources, University of Michigan, and Princeton University.

In the private sector, an equal number of major national organizations purchased during the same period.

You can call A.E.I. too!

*With our normal fast shipping, we have time to test; expedited shipment may preclude testing.
**A.E.I. does not wish to imply that any of these fine organizations endorse A.E.I., merely that A.E.I. is proud to have them as customers.



S.S.M.	KIT	ASM
CB2 Z80 CPU	$200	$265
VB1C VIDEO	140	190
VB2 VIDEO	155	210
VB3 VIDEO	375	440
IO4 INTERFACE	165	225
SB1 SYNTHESIZER	195	270

MEASUREMENT SYSTEMS MEMORY

DM3200 32K 4MHZ	480
DM6400 64K 4MHZ	595
DMB3200 32K 4MHZ BANK SELECT	630
DMB6400 64K 4MHZ BANK SELECT	745

DYNABYTE SYSTEMS ALSO AVAILABLE—CALL FOR THE EXCITING NEWS ABOUT NEW COMPETITIVE PRICING!

WE WILL TRY TO BEAT ANY ADVERTISED PRICE.

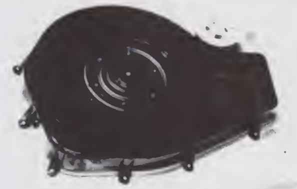
TERMS: All prices listed are cash discounted and subject to change. COD's are 2% additional. Universities and well-rated firms Net-10.

SHIPPING: ADDITIONAL IN ALL CASES.

TECHNICAL ASSISTANCE:
WE TRY TO HELP INTERFACE AND TROUBLESHOOT—CALL (714) 963-1414.

Open Monday-Friday, 8:00 a.m.-5:00 p.m.

Dealer Inquiries Invited.



MORROW

DECISION 1 BASIC	$1550
DISCUS-1 2 DRIVE	1550
DISCUS 2D 1 DRIVE	935
DISCUS 2D 2 DRIVE	1550
DISCUS 2+2 1 DRIVE	1250
DISCUS 2+2 2 DRIVE	2190
DISC JOCKEY 2D	350
DISCUS M26 HARD DISC	3925
ADDITIONAL HARD DISC	3685
DISCUS *M10*	2990

MISCELLANEOUS

NEC THIMBLES	$16
NEC RIBBONS	7
RS2332 CABLE 5'	20
RS 232 CABLE 10"	25
LEDEX MONITOR	145
NOVATION CAT	145

SOFTWARE

WORDSTAR	$320
MAILMERGE	100
DATASTAR	250
N.S. PASCAL	295
MAGIC WAND	290
C BASIC	100

GRAHAM-DORIAN

JOB COSTING	$700
INVENTORY	475
CASH REG.	475
APARTMENT	475
MEDICAL	700

ADDITIONAL SOFTWARE DISCOUNTS WITH SYSTEM PURCHASE

DISCS—BOX OF 10

VERBATUM 5¼ 1 SIDE	$27
VERBATUM 5¼ 2 SIDE	45
VERBATUM 8 1 SIDE	35
VERBATUM 8 2 SIDE	55
OTHERS	CALL
PLASTIC STORAGE BOXES	3

OUR SYSTEMS WORK!

All systems normally tested and configured in our repair facility before delivery. Service contracts available. Prompt repairs and warranties.

CLUB NOTES

Microcomputer Investment Management

The MicroComputer Investors Association is a nonprofit group of professionals who use microcomputers in managing investments. Their publication, *The MicroComputer Investor*, contains programs for investors who use microcomputers. A membership application and an index of all programs and articles published to date is available for $3 from Jack Williams, MCIA, 902 Anderson Drive, Fredericksburg, VA 22401.

Computerfest '81

The Midwest Affiliation of Computer Clubs (MACC) is sponsoring Computerfest '81 at Franklin University, Columbus, OH, on May 29–31. Seminars and workshops, as well as exhibits, a flea market and contests, will be featured. For more information, send an SASE to Computerfest '81, c/o Paul Pittenger, 215 Delphi, Apt. J, Columbus, OH 43202.

Southeastern Michigan Computer Organization

The Southeastern Michigan Computer Organization (SEMCO) holds its meetings on the second Sunday of each month at the Ford Automotive Safety Center Auditorium in Detroit at 7 PM. All interested persons are welcome to attend. For further information, write to SEMCO, Box 02426, Detroit, MI 48202.

Technico User's Group

A Technico User's Group (TUG) has recently been formed for users of Technico computers or other computers based on TI 990 or TMS9900 architecture. The group plans to publish a newsletter and is soliciting articles on software, hardware and applications, as well as ads for software and hardware for sale. Contact Penn S. Avera, Quantum Data Systems, Inc., 259 So. Farragut Ter., Philadelphia, PA 19139.

COMPUTER CLINIC

We recently purchased a Mits Altair 8800-B Turnkey with 5-1/4 inch disks. The system came with BASIC, Release 5.0, and DOS manuals, but did not come with the DOS software. We need some help in obtaining information on the location and price of the DOS software. Also, we would be interested in any documentation on the various computer boards used by this system.

Chris E. Lyons
Vice-president
Hadley Enterprises, Inc.
16019 N. 68th Ave.
Peoria, AZ 85345

I have purchased a used SOL 8080 Processor Tech computer and a bare 16K–64K Dynamic RAM board. I have no schematics or parts listing for this board. Can anyone provide me with this information?

Keith A. Mansfield
1272 Elmsmere Lane
Riverdale, GA 30296

Does anyone have the KIM text editor/KIM assembler manual and listing published by MOS Technology Sales in January 1977? I need a copy of the editor/assembler hex dump.

Earl T. Brant III
PO Box 74
Dorothy, NJ 08317

CALENDAR

Trenton Computer Festival

The sixth annual Trenton Computer Festival will be held April 25–26 at Trenton State College. It will feature a flea market, commercial exhibits, speakers, user group sessions, free short courses, demonstrations and a computer graphics theater. Computer conference sessions and forums will be held on microcomputers in the home, education, medicine, amateur radio, music and the arts. Contact Dr. Allen Katz, Trenton State College, Hillwood Lakes, PO Box 940, Trenton, NJ 08625, 609-771-2487.

Europe Software 1981

Europe Software 1981 will run from June 2–4 in Utrecht, The Netherlands. This international exhibition is billed as a promotion for the transfer of information about computer software products, software services, bureau services and education in information science. Inquiries should be addressed to Royal Netherlands Industries Fair, PO Box 8500, 3503 RM Utrecht, The Netherlands. Telephone: (30) 914 914, Telex: 47132.

Computers and the Humanities

The University of Michigan will host the Fifth International Conference on Computers and the Humanities (ICCH/5) on May 17–20 and the Joint Conference on Easier and More Productive Use of Computing Systems on May 20–22. For further information, contact Ida M. Sanburn, 4258 Institute for Social Research, University of Michigan, Ann Arbor, MI 48106.

Electro/81

The Electro Exhibition and Convention, a salute to advances in electronics products and ideas, will be held April 7–9 in New York City. Exhibits, including a special display of small computers for engineering and business applications, will be held at the New York Coliseum. The program schedule, to be held at the Sheraton Centre Hotel, will include technical sessions on microprocessor technology. Also to be featured are home and personal computers, instrumentation and peripheral concepts, analog interface and microcomputers. Contact Electro, 999 North Sepulveda Boulevard, El Segundo, CA 90245.

National TRS-80 Microcomputer Show

The First Annual National TRS-80 Microcomputer Show will be held in New York City on May 22, 23 and 24 at the New York Statler Exposition Center (opposite Madison Square Garden and Pennsylvania Railroad Station). The show will feature commercial exhibits, seminars and speakers. For further information, contact Kengore Corporation, 3001 Route 27, Franklin Park, NJ 08823, 201-297-6918.

Federal DP Expo

Federal DP Expo, the seventh annual conference and exposition for computer system users in the U. S. Government, will be held April 14–16 at the Sheraton Washington Hotel in Washington, DC. More than 150 computer-related companies will display and demonstrate hardware/software products, systems and services. Contact The Interface Group, 160 Speen St., Framingham, MA 01701, or call 800-225-4620; in Massachusetts call 617-879-4502.

Computers in Education

The Third Annual Computers in Education Conference will be held at Seattle Pacific University, Seattle, WA, on May 1–2. It will feature panel discussions, talks, workshops and exhibits, with a special emphasis on the use of microcomputers in grades K–12. For further information, contact Jerry Johnson, Seattle Pacific University, Seattle, WA 98119.

Southwest Computers Show

The Southwest Computer Show & Office Equipment Exposition will be held in Market Hall at the Dallas Market Center, April 9–12. The exposition is produced by National Computer Shows, 824 Boylston St., Chestnut Hill, MA 02167, 617-739-2000.

Amateur Fair

The North Area Repeater Association will sponsor the state's largest swapfest and exposition for computer hobbyists and radio amateurs on May 30 at the Minnesota State Fairgrounds in St. Paul. Exhibits, booths and prizes will be featured. For information or reservations, write Amateur Fair, PO Box 30054, St. Paul, MN 55175.

National Computer Conference

The National Computer Conference is scheduled for May 4–7, in McCormick Place, Chicago. Exhibits, program sessions and seminars, as well as special activities, are planned. For further information, contact AFIPS, PO Box 9658, 1815 N. Lynn St., Arlington, VA 22209, 703-558-3610.

DEALER DIRECTORY

Phoenix, AZ
Software Specialists! Authorized dealers in quality software packages. Experts provide comprehensive product demonstrations, outstanding support and free consultation. Resident CP/M swami. Arizona's finest service department. **The Denver-Ridge Group, Suite 104, 8900 North Central Avenue, Phoenix, AZ 85020, 944-8371.**

El Monte, CA
Ohio Scientific specialist in the San Gabriel Valley serving greater Los Angeles. Full product line on display. Specializing in business computers. In-house service. Custom programming. Terminals. Printers. Open Mon–Sat, 9 AM–7 PM. **Computer & Video, 3380 Flair Dr., Suite 207, El Monte, CA 91731, 572-7292.**

San Jose, CA
Bay area's newest computer software store. Featuring Instant Software for the TRS-80, magazines, books. **Shaver Radio, 1378 S. Bascom Ave., San Jose, CA 95128, 998-1103.**

San Jose, CA
New and used computer products—specializing in S-100 boards, printers, drives, chassis and complete systems, as well as supplies and parts—Imsai, Tandon, Diablo—5000 sq. ft. **W/W Component Supply, Inc., 1771 Junction Ave., San Jose, CA 95112.**

Sarasota, FL
Dynabyte computer systems, Hazeltine and NEC, Word-Star, Structured Systems accounting. Consulting, training, sales, service. **Glisco, Inc., 4001 Roberts Point Rd., Sarasota, FL 33581, 349-0200.**

Sarasota, FL
Real-Time/Industrial Software Engineering Consultants. Top-down design and structured techniques utilized. **Advanced Computer Concepts, Inc., PO Box 15710, Sarasota, FL 33579.**

Tampa, FL
Apple Computer sales and service. S-100 boards from SSM, Godbout, Thinker Toys, California Computer Systems. Computer books and magazines. **AMF Microcomputer Center, Inc., 11158 N. 30th Street, Tampa, FL 33612, 971-4072, 977-0708.**

Aurora, IL
Microcomputer systems for home or business; peripherals, software, books & magazines. Apple, Hewlett-Packard Series 80 Systems, HP Calculators, IDS, Qume, Starwriter printers. **Farnsworth Computer Center, 1891 N. Farnsworth Ave., Aurora, IL 60505, 851-3888.**

Chicago, IL
Computer Hardware/Specialists for home and business. Largest selection of computer books, magazines and copyrighted software in Illinois. Experienced factory-trained service department. Feature Apple, Alpha Microsystems and Hewlett-Packard calculators and accessories. **Data Domain of Schaumburg, 1612 E. Algonquin Rd., Schaumburg, IL 60195, 397-8700.**

Chicago, IL
Brand new lowest prices, never undersold, postpaid in USA—Teletype 43 keyboard printers, Okidata & Integral Data printers, SS-50 bus computers, peripherals & business software. **Data Mart, 914 East Waverly Street, Arlington Heights, IL 60004, 398-8525.**

Herington, KS
Hardware support. Maintenance and service for all microcomputers and peripherals. Kits assembled or debugged. Radio Shack (mods OK) repaired. Quality work, fast turnaround and reasonable cost. **Prairie Micro Clinic, Box 325, Herington, KS 67449, 258-2179.**

Call or visit the dealer nearest you!

East Lansing, MI
Computers by Altos, APF, Atari, Exidy, Intertec, Microdasys, Ohio Scientific, Zenith, accessories (printers, disk), software (customized, packaged), supplies (disks, ribbons, paper), training. One stop shopping. **Wizards Arsenal, 4964 Northwind Drive, East Lansing, MI 48823, 337-7892.**

Garden City, MI
Books, magazines, hardware and software for Apple, North Star, TRS-80 and PET. **Computer Center, 28251 Ford Rd., Garden City, MI 48135, 425-2470.**

Hannibal, MO
Ohio Scientific products, modifications, service, software. 8" disk for C1P, C4P. Process control specialist. **E&I Technical Service, 5300 Paris Gravel Road, Hannibal, MO 63401, 248-0084.**

Long Island, NY
Software-hardware application engineering. System startups & tutorial sessions. Program customizing, translation & development. Data communication networks. World-wide service available. Local experts needed. **Charles Stewart, 442 Lockwood Dr., No. Shirley, NY 11967, 924-5164.**

Portland, OR
Ohio Scientific specialists for business and personal computers. Local service. Terminals, printers, custom programming. Full OSI product line on display! 10 AM to 6 PM M–F. **Fial Computer, 11266 SE 21st Ave., Milwaukie, OR 97222, 654-9574.**

Dealers: Listings are **$15** per month in prepaid quarterly payments, or one yearly payment of **$150**, also prepaid. Ads include 25 words describing your products and services plus your company name, address and phone. (No area codes or merchandise prices, please.) Call Marcia at 603-924-7138 or write *Kilobaud Microcomputing*, Ad Department, Peterborough, NH 03458.

New Sensibly-Priced Paper Tape Reader

A new, smaller, less expensive Stand-Alone Paper Tape Reader Model 605 reads at 150 CPS, has a parallel TTL output and is bi-directional. It stops on character, is completely self-contained and has automatic taut-tape sensing.
The new reader has 5 volt DC and 24 volt DC output power available and an optional internal clock. The desk-top model is priced at $495 for one unit, a saving of about 25% of the price of the larger Model 612.
Contact Louis Doshay, Addmaster Corporation, 416 Junipero Serra Drive, San Gabriel 91776; telephone (213) 285-1121 ✓261

Subscription Problem?

Kilobaud Microcomputing does not keep subscription records on the premises, therefore calling us only adds time and doesn't solve the problem.

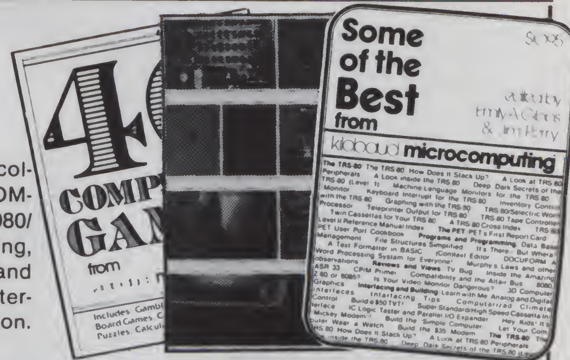
Please send a description of the problem and your most recent address label to:

Kilobaud Microcomputing
Subscription Dept.
PO Box 997
Farmingdale, NY 11737

Thank you and enjoy your subscription.

kb microcomputing book nook

3 NEW BOOKS

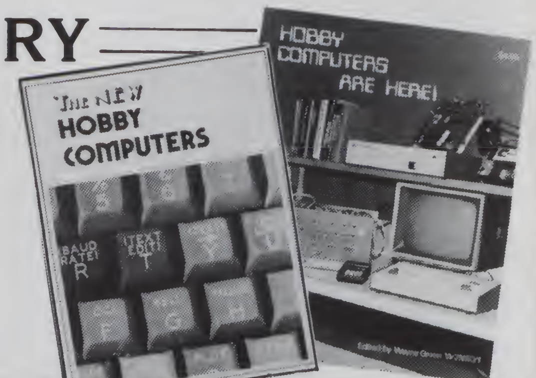


● **SOME OF THE BEST FROM KILOBAUD/MICROCOMPUTING**—BK7311—A collection of the best articles that have recently appeared in Kilobaud/MICROCOMPUTING. Included is material on the TRS-80 and PET systems, CP/M, the 8080/8085/Z80 chips, the ASR-33 terminal. Data base management, word processing, text editors and file structures are covered too. Programming techniques and hardcore hardware construction projects for modems, high speed cassette interfaces and TVTs are also included in this large format, 200 plus page edition. $10.95.*

● **UNDERSTANDING AND PROGRAMMING MICROCOMPUTERS**—BK7382—A valuable addition to your computing library. This two part text includes the best articles that have appeared in 73 and Kilobaud Microcomputing magazines on the hardware and software aspects of the new microcomputing hobby. Well known authors and well structured text helps the reader get involved in America's fastest growing hobby. $10.95*

● **40 COMPUTER GAMES**—BK7381—Forty games in all in nine different categories. Games for large and small systems, and even a section on calculator games. Many versions of BASIC used and a wide variety of systems represented. A must for the serious computer gamesman. $7.95*

INTRODUCTORY



● **HOBBY COMPUTERS ARE HERE!**—BK7322—If you (or a friend) want to come up to speed on how computers work . . . hardware and software . . . this is an excellent book. It starts with the fundamentals and explains the circuits, and the basics of programming. This book has the highest recommendations as a teaching aid for newcomers. $4.95.*

● **THE NEW HOBBY COMPUTERS**—BK7340—This book takes it from where "HOBBY COMPUTERS ARE HERE!" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards . . . and much, much more! Don't miss this tremendous value! Only $4.95.*

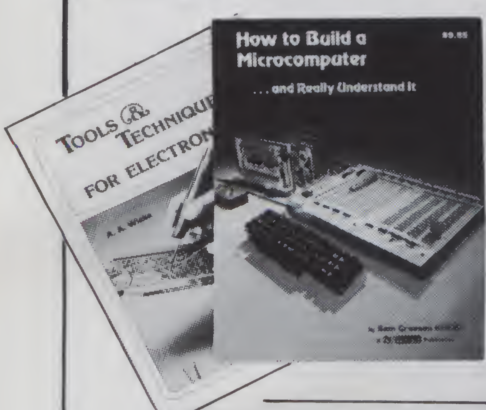
INTRODUCTION TO MICROCOMPUTERS (VOL. 0→III)

● **AN INTRODUCTION TO MICROCOMPUTERS, VOL. 0**—BK1130—The Beginner's Book—Written for readers who know nothing about computers—for those who have an interest in how to use computers—and for everyone else who must live with computers and should know a little about them. The first in a series of 4 volumes, this book will explain how computers work and what they can do. Computers have become an integral part of life and society. During any given day you are affected by computers, so start learning more about them with Volume 0. $7.95.*

● **VOL. I**—BK1030—**2nd Edition completely revised.** Dedicated to the basic concepts of microcomputers and hardware theory. The purpose of Volume I is to give you a thorough understanding of what microcomputers are. From basic concepts (which are covered in detail), Volume I builds the necessary components of a microcomputer system. This book highlights the difference between minicomputers and microcomputers. $12.99.*

● **VOL. II**—BK1040 (with binder)—Contains descriptions of individual microprocessors and support devices used only with the parent microprocessor. Volume II describes all available chips. $31.99*

● **VOL. III**—BK1133 (with binder)—Contains descriptions of all support devices that can be used with any microprocessor. $21.99*



● **HOW TO BUILD A MICROCOMPUTER—AND REALLY UNDERSTAND IT**—BK7325—by Sam Creason. The electronics hobbyist who wants to build his own microcomputer system now has a practical "How-To" guidebook. This book is a combination technical manual and programming guide that takes the hobbyist step-by-step through the design, construction, testing and debugging of a complete microcomputer system. Must reading for anyone desiring a true understanding of small computer systems. $9.95.*

● **TOOLS & TECHNIQUES FOR ELECTRONICS**—BK7348—by A. A. Wicks is an easy-to-understand book written for the beginning kit builder as well as the experienced hobbyist. It has numerous pictures and descriptions of the safe and correct ways to use basic and specialized tools for electronic projects as well as specialized metal working tools and the chemical aids which are used in repair shops. $4.95.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

"PRICES SUBJECT TO CHANGE WITHOUT NOTICE"

FOR TOLL FREE ORDERING CALL 1-800-258-5473

kb microcomputing book nook

SPECIAL INTERESTS

new!

new!

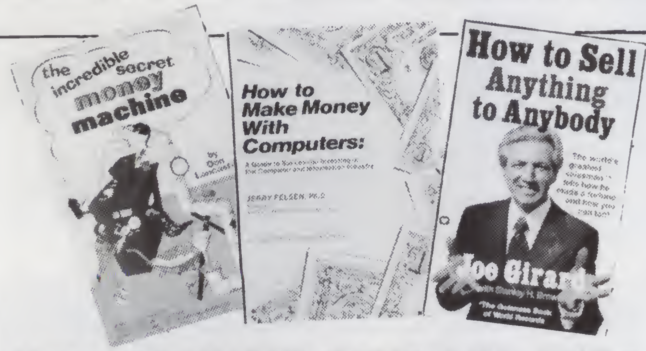
●**MICROSOFT BASIC DECODED AND OTHER MYSTERIES**—BK1186—by James Farvour. From the company that brought you *TRS-80 DISK AND OTHER MYSTERIES*! Contains more than 6500 lines of comments for the disassembled Level II ROMs, six additional chapters describing every BASIC subroutine, with assembly language routines showing how to use them. Flow charts for all major routines give the reader a real insight into how the interpreter works. $29.50 (Available after December 20th).

●**THE CP/M HANDBOOK (with MP/M)**—BK1187—by Rodnay Zaks. A complete guide and reference handbook for CP/M—the industry standard in operating systems. Step-by-step instruction for everything from turning on the system and inserting the diskette to correct user discipline and remedial action for problem situations. This also includes a complete discussion of all versions of CP/M up to and including 2.2, MP/M and CDOS. $14.95*.

●**TRS-80 DISK AND OTHER MYSTERIES**—BK1181—by Harvard C. Pennington. This is the definitive work on the TRS-80 disk system. It is full of detailed "How to" information with examples, samples and in-depth explanations suitable for beginners and professionals alike. The recovery of one lost file is worth the price alone. $22.50.*

●**INTRODUCTION TO TRS-80 GRAPHICS**—BK1180—by Don Inman. Dissatisfied with your Level I or Level II manual's coverage of graphics capabilities? This well-structured book (suitable for classroom use) is ideal for those who want to use all the graphics capabilities built into the TRS-80. A tutorial method is used with many demonstrations. It is based on the Level I, but all material is suitable for Level II use. $8.95.*

●**MICROPROCESSOR INTERFACING TECHNIQUES**—BK1037—by Austin Lesea & Rodnay Zaks—will teach you how to interconnect a complete system and interface it to all the usual peripherals. It covers hardware and software skills and techniques, including the use and design of model buses such as the IEEE 488 or S100. $15.95.*



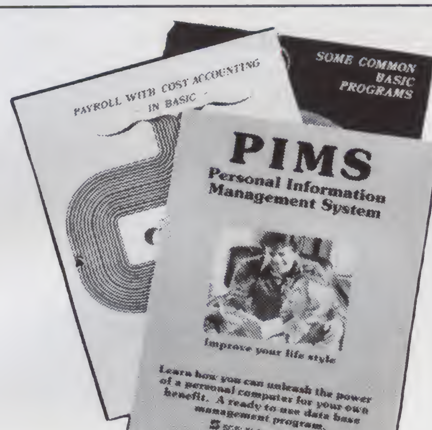
MONEYMAKING

●**HOW TO MAKE MONEY WITH COMPUTERS**—BK1003—In 10 information-packed chapters, Jerry Felsen describes more than 30 computer-related, money-making, high profit, low capital investment opportunities. $15.00.*

●**HOW TO SELL ANYTHING TO ANYBODY**—BK7306—According to *The Guinness Book of World Records*, the author, Joe Girard, is "the world's greatest salesman." This book reveals how he made a fortune—and how you can, too. $2.25.*

●**THE INCREDIBLE SECRET MONEY MACHINE**—BK1178—by Don Lancaster. A different kind of "cookbook" from Don Lancaster. Want to slash taxes? Get free vacations? Win at investments? Make money from something that you *like* to do? You'll find this book essential to give you the key insider details of what is really involved in starting up your own money machine. *$5.95.**

BUSINESS



●**PAYROLL WITH COST ACCOUNTING—IN BASIC**—BK1001—by L. Poole & M. Borchers, includes program listings with remarks, descriptions, discussions of the principle behind each program, file layouts, and a complete user's manual with step-by-step instructions, flowcharts, and simple reports and CRT displays. Payroll and cost accounting features include separate payrolls for up to 10 companies, time-tested interactive data entry, easy correction of data entry errors, job costing (labor of distribution), check printing with full deduction and pay detail, and 16 different printed reports, including W-2 and 941 (in **CBASIC**). $20.00.*

●**SOME COMMON BASIC PROGRAMS**—BK1053—published by Adam Osborne & Associates, Inc. Perfect for non-technical computerists requiring ready-to-use programs. Business programs, plus miscellaneous programs. Invaluable for the user who is not an experienced programmer. All will operate in the stand-alone mode. $14.99 paperback.

●**PIMS: PERSONAL INFORMATION MANAGEMENT SYSTEM**—BK1009—Learn how to unleash the power of a personal computer for your own benefit in this ready-to-use data-base management program. $11.95.*

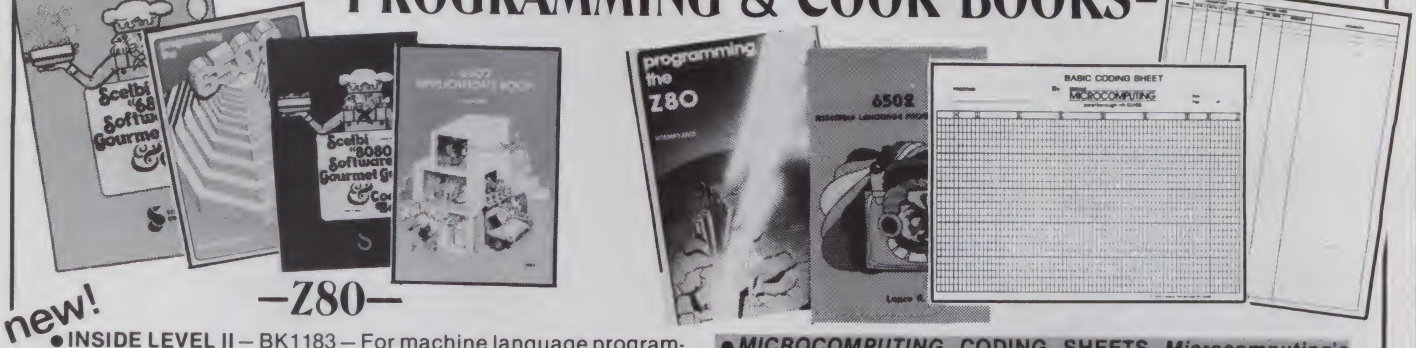
"PRICES SUBJECT TO CHANGE WITHOUT NOTICE"

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

kb microcomputing book nook

—PROGRAMMING & COOK BOOKS—



—Z80—

new!

● **INSIDE LEVEL II** — BK1183 — For machine language programmers! This is a comprehensive reference guide to the Level II ROMs, allowing easy utilization of the sophisticated routines they contain. It concisely explains set-ups, calling sequences, variable passage and I/O routines. Part II presents an entirely new composite program structure which unloads under the SYSTEM command and executes in both BASIC and machine code with the speed and efficiency of a compiler. Special consideration is given to disk systems. $15.95.*

● **PROGRAMMING THE Z-80** — BK1122 — by Rodnay Zaks. Here is assembly language programming for the Z-80 presented as a progressive, step-by-step course. This book is both an educational text *and* a self-contained reference book, useful to both the beginning and the experienced programmer who wish to learn about the Z-80. Exercises to test the reader are included. $14.95.*

● **Z-80 ASSEMBLY LANGUAGE PROGRAMMING** — BK1177 — by Lance A. Leventhal. This book thoroughly covers the Z80 instruction set, abounding in simple programming examples which illustrate software development concepts and actual assembly language usage. Features include Z80 I/O devices and interfacing methods, assembler conventions, and comparisons with 8080A/8085 instruction sets and interrupt structure. $16.99.*

● **Z-80 SOFTWARE GOURMET GUIDE AND COOKBOOK** — BK1045 — by Nat Wadsworth. Scelbi's newest cookbook! This book contains a complete description of the powerful Z-80 instruction set and a wide variety of programming information. Use the author's ingredients including routines, subroutines and short programs, choose a time-tested recipe and start cooking! $15.95.*

—6502—

● **PROGRAMMING THE 6502 (Third Edition)**—BK1005— Rodnay Zaks has designed a self-contained text to learn programming, using the 6502. It can be used by a person who has never programmed before, and should be of value to anyone using the 6502. The many exercises will allow you to test yourself and practice the concepts presented. $12.95.*

● **6502 APPLICATIONS BOOK** — BK1006 — Rodnay Zaks presents practical-application techniques for the 6502 microprocessor, assuming an elementary knowledge of microprocessor programming. You will build and design your own domestic-use systems and peripherals. Self-test exercises included. $12.95.*

● **6502 ASSEMBLY LANGUAGE PROGRAMMING**—BK1176— by Lance A. Leventhal. This book provides comprehensive coverage of the 6502 microprocessor assembly language. Leventhal covers over 80 programming examples from simple memory load loops to complete design projects. Features include 6502 assembler conventions, input/output devices and interfacing methods, and programming the 6502 interrupt system. $16.99.*

● **6502 SOFTWARE GOURMET GUIDE AND COOKBOOK** — BK1055 — by Robert Findley. This book introduces the BASIC language programmer into the realm of machine-language programming. The description of the 6502 structure and instruction set, various routines, subroutines and programs are the ingredients in this cookbook. "Recipes" are included to help you put together exactly the programs to suit your taste. $12.95.*

● ***MICROCOMPUTING* CODING SHEETS** *Microcomputing*'s dozen or so programmers wouldn't try to work without these handy scratch pads, which help prevent the little errors that can cost hours and hours of programming time. Available for programming is Assembly/Machine Language (PD1001), which has columns for address, instruction (3 bytes), source code (label, op code, operand) and comments; and for BASIC (PD1002) which is 72 columns wide. 50 sheets to a pad. $2.39.*

—8080 / 8080A—

● **8080A/8085 Assembly Language Programming**—by Lance Leventhal—BK1004—Assembly language programming for the 8080A/8085 is explained with a description of the functions of assemblers and assembly instructions, and a discussion of basic software development concepts. Many fully debugged, practical programs are included as is a special section on structured programming. $15.99.*

● **8080 PROGRAMMING FOR LOGIC DESIGN**—BK1078—Ideal reference for an indepth understanding of the 8080 processor. Application-oriented and the 8080 is discussed in light of replacing conventional, hard-wired logic. Practical design considerations are provided for the implementation of an 8080-base control system. $9.50.*

● **8080 SOFTWARE GOURMET GUIDE AND COOKBOOK** — BK1102 — If yu have been spending too much time developing simple routines for your 8080, try this new book by Scelbi Computing and Robert Findley. Describes sorting, searching, and many other routines for the 8080 user. $12.95.*

—6800—

● **6800 PROGRAMMING FOR LOGIC DESIGN**—BK1077—Oriented toward the industrial user, this book describes the process by which conventional logic can be replaced by a 6800 microprocessor. Provides practical information that allows an experimenter to design a complete micro control system from the "ground up." $9.50.*

● **6800 SOFTWARE GOURMET GUIDE AND COOKBOOK** — BK1075 — Like its culinary cousin, *The 8080 Gourmet Guide*, this book by Scelbi Computing and Robert Findley describes sorting, searching and other routines — this time for the 6800 user. $12.95.*

—COOK BOOKS—

● **CMOS COOKBOOK** — BK1011 — by Don Lancaster. Details the application of CMOS, the low power logic family suitable for most applications presently dominated by TTL. Required reading for every serious digital experimenter! $10.50.*

● **TVT COOKBOOK** — BK1064 — by Don Lancaster. Describes the use of a standard television receiver as a microprocessor CRT terminal. Explains and describes character generation, cursor control and interface information in typical, easy-to-understand Lancaster style. $9.95.*

● **TTL COOKBOOK** — BK1063 — by Donald Lancaster. Explains what TTL is, how it works, and how to use it. Discusses practical applications, such as a digital counter and display system, events counter, electronic stopwatch, digital voltmeter and a digital tachometer. $9.50.*

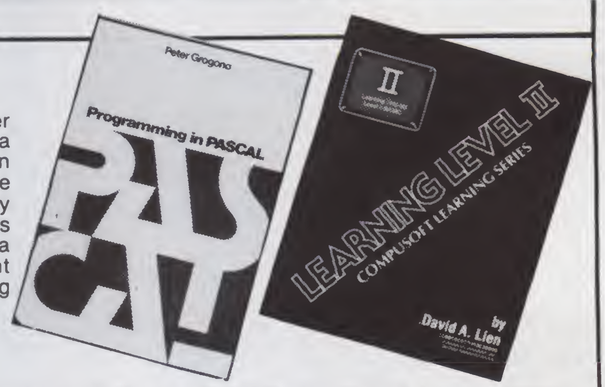
"PRICES SUBJECT TO CHANGE WITHOUT NOTICE"

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

kb microcomputing book nook

BASIC AND PASCAL



NEW REVISED EDITION

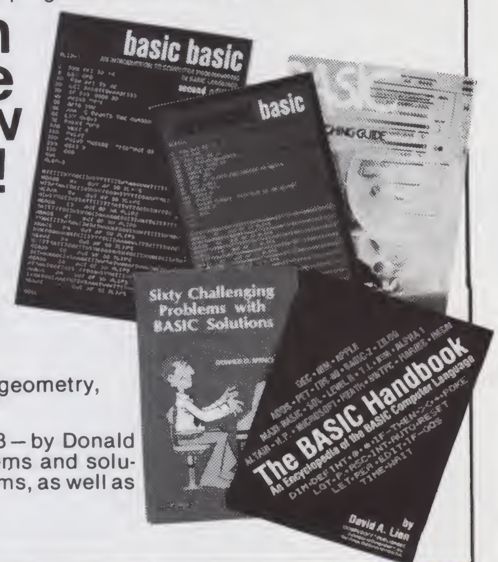
● **PROGRAMMING IN PASCAL**—BK1140—by Peter Grogono. The computer programming language PASCAL was the first language to embody in a coherent way the concepts of structured programming, which has been defined by Edsger Dijkstra and C.A.R. Hoare. As such, it is a landmark in the development of programming languages. PASCAL was developed by Niklaus Wirth in Zurich; it is derived from the language ALGOL 60 but is more powerful and easier to use. PASCAL is now widely accepted as a useful language that can be efficiently implemented, and as an excellent teaching tool. It does not assume knowledge of any other programming language; it is therefore suitable for an introductory course. $12.95.*

● **THE BASIC HANDBOOK**—BK1174—by David Lien. This book is unique. It is a virtual ENCYCLOPEDIA of BASIC. While not favoring one computer over another, it explains over 250 BASIC words, how to use them and alternate strategies. If a computer does not possess the capabilities of a needed or specified word, there are often ways to accomplish the same function by using another word or combination of words. That's where the HANDBOOK comes in. It helps you get the most from your computer, be it a "bottom-of-the-line" micro or an oversized monster. $14.95.*

● **LEARNING LEVEL II**—BK1175—by David Lien. Written especially for the TRS-80, this book concentrates on Level II BASIC, exploring every important BASIC language capability. Updates are included for those who have studied the Level I User's Manual. Sections include: how to use the Editor, dual cassette operation, printers and peripheral devices, and the conversion of Level I programs to Level II. $15.95.*

● **INTRODUCTION TO PASCAL**—BK1189—by Rodnay Zaks. A step-by-step introduction for anyone wanting to learn the language quickly and completely. Each concept is explained simply and in a logical order. All features of the language are presented in a clear, easy-to-understand format with exercises to test the reader at the end of each chapter. It describes both standard PASCAL and UCSD PASCAL, the most widely used dialect for small computers. No computer or programming experience is necessary. $14.95.*

new!



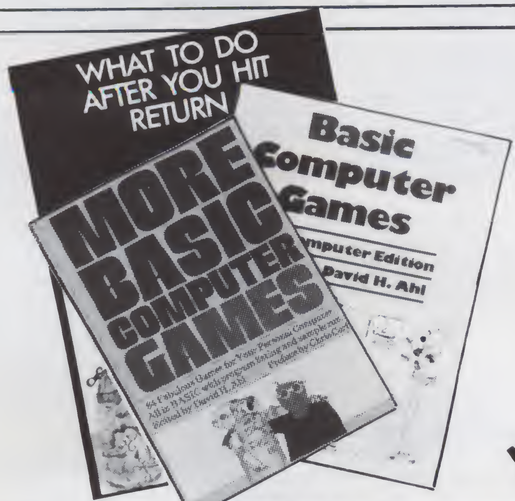
● **BASIC NEW 2ND EDITION**—BK1081—by Bob Albrecht. Self-teaching guide to the computer language you will need to know for use with your microcomputer. This is one of the easiest ways to learn computer programming. $6.95.*

● **BASIC BASIC (2ND EDITION)**—BK1026—by James S. Coan. This is a textbook which incorporates the learning of computer programming using the BASIC language with the teaching of mathematics. Over 100 sample programs illustrate the techniques of the BASIC language and every section is followed by practical problems. This second edition covers character string handling and the use of data files. $10.50.*

● **ADVANCED BASIC**—BK1000—Applications, including strings and files, coordinate geometry, area, sequences and series, simulation, graphing and games. $10.75*.

● **SIXTY CHALLENGING PROBLEMS WITH BASIC SOLUTIONS (2nd Edition)**—BK1073—by Donald Spencer, provides the serious student of BASIC programming with interesting problems and solutions. No knowledge of math above algebra required. Includes a number of game programs, as well as programs for financial interest, conversions and numeric manipulations. $6.95.*

GAMES



new!

● **WHAT TO DO AFTER YOU HIT RETURN**—BK1071—PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions, many illustrations. Lunar Landing, Hammurabi, King, Civel 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. $10.95.*

● **BASIC COMPUTER GAMES**—BK1074—Okay, so once you get your computer and are running in BASIC, then *what? Then* you need some programs in BASIC, that's what. This book has 101 games for you from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. $7.50.*

● **MORE BASIC COMPUTER GAMES**—BK1182—edited by David H. Ahl. More fun in BASIC! 84 new games from the people who brought you *BASIC Computer Games.* Includes such favorites as Minotaur (battle the mythical beast) and Eliza (unload your troubles on the doctor at bargain rates). Complete with game description, listing and sample run. $7.50.*

"PRICES SUBJECT TO CHANGE WITHOUT NOTICE"

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to Kilobaud *Microcomputing* Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information. No C.O.D. orders accepted. All orders add $1.00 handling. Please allow 4-6 weeks for delivery. Questions regarding your order? Please write to Customer Service at the above address.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

CLASSIFIEDS

Classified advertisements are intended for use by persons desiring to buy, sell or trade used computer equipment. No commercial ads are accepted.

Two sizes of ads are available. The $5 box allows up to 5 lines of about 35 characters per line, including spaces and punctuation. The $10 box allows up to 10 lines. Minimize use of capital letters to save space. No special layouts allowed. Payment is required in advance with ad copy. We cannot bill or accept credit.

Advertising text and payment must reach us 60 days in advance of publication (i.e., copy for March issue, mailed in February, must be here by Jan. 1). The publisher reserves the right to refuse questionable or inapplicable advertisements. Mail copy with payment to: **Classifieds, Kilobaud Microcomputing, Peterborough NH 03458**. Do not include any other material with your ad as it may be delayed.

Commodore PET. 16K RAM expandable to 32K on-board. 14K ROM w/machine-language monitor & improved BASIC. 8K ROM expansion sockets on-board. Typewriter keyboard and number pad. C2N external cassette recorder. Technical manuals plus "PET & IEEE Bus" and "PET Personal Computer Guide" books. $750, plus shipping. Greg after 5 PM at 513-851-1936.

North Star 2 system. Double density, 48K, with D.C. Hayes modem. 2 serial & 1 parallel ports, Soroc IQ 140 terminal, Centronics 779 tractor printer, includes apx $2000 in business software, 6 months old, in org boxes, $4895 or best offer. Also one D.C. Hayes modem 100, $325. Call Don, 615-526-7651.

Software (personal) for sale. Sold system must sell at super reduced prices. Dozens of CP/M, languages, business s/w (G/L, A/R, A/P, etc.). Send SASE for complete list and prices. Dee K. Moser, Box 638, Great Bend, KS 67530.

Super electronics shop for $800! Instruments, texts (all Kilobaud natch), components, etc., everything. Owner can't use the approx. $1800 tax credit. A roomful of prime at choice price! SASE for inv. $1 for pix. 334 Riverside Drive, Palm Beach Gardens, FL 33410.

For Sale: Magic Wand disk and manual, $200. I have switched to Scripsit. A. A. Schwartz, 6454 El Camino Teatro, La Jolla, CA 92037.

For Sale: Altair 8800 with 32K RAM, 88-2SIO, 88-ACR, Microsoft Extended BASIC in ROM, complete documentation, factory-assembled chassis: $1000, plus shipping. ASR-33 Teletype with tape reader/punch, stand and manuals, factory-reconditioned: $400, plus shipping. Gene Szymanski, 693 Rosedale Road, Princeton, NJ 08540. Phone 609-924-8856, evenings.

SYM w/4K RAM and extra 6522 chip for full I/O capability. Three manuals included. MTU five slot motherboard and technical manuals for high resolution video board, PROM I/O board and others. $200, plus shipping. Call Greg after 5 PM EST 513-851-1936.

SWTPC 4K mem. brd., $45; 8K mem. brd., $95; JPC CK-7 RT clock w/aux. pwr., $45; MicroWare RT/68 ROM, $45; Ex. Cond./full docs.: C. R. Silvia, PO Box 234, Hines, IL 60141.

For Sale: Assembled, operational; Heath H8 w/H8-4 four-port serial board, full 64K Trionix memory (150 nsec 4116s), PAMGO ROM, dual drive H8-17 w/HDOS, MBASIC, many HUG and other disks, H19 terminal, covers for all. First $2000 or best offer. I'll ship. Two 8K boards—make offer. Eric Sturm, Mountain View Mobile Park, Lot B-4, Denver, CO 80230. 303-399-6867.

For Sale: One Houston Instruments 11" by 11" bitpad. James Kientz, RR #1, Wamego, KS 66547. Home 913-532-3722.

Have purchased a mini. Selling all my micro equipment and lots of software for CP/M and North Star with it. Send SASE for a list of equipment and software. RK&B, Inc., PO Box 178, Kechi, KS 67067.

CPU & 2 32K RAM boards for Heath H8. Build a 64K computer for less than current prices. $595 or will sell boards separately. W. S. Hall, 5621 Maple Heights, Pittsburgh, PA 15232.

Wanted: Information where I can find the King James version of the Bible in computer readable format on disk or cassette. Steven Tilden, 4771 S. Warren Ave., Tucson, AZ 85714. 602-746-0569. I will accept collect calls if you have this information.

Memory boards from bankruptcy sale. Limited quantity of 16K dynamic RAMs. Configured for 18 bits, 8K words or 9 bits, 16K bytes, on board refresh & add. to 20 bits. Schematics, interface information & 110-pin gold plated, wire-wrap connector. Functionally tested 1-3 @ $100, 3-5 @ $85. Untested and others from $40. TMI, 2000 N. Highway 67, Florissant, MO 63033. 314-837-8338.

---

4 Sale: SSB BFD-68 disk controller board, with 5" mini DOS, editor, assembler, text processor & 3 versions of disk BASIC. $200. Call after 4 PM 414-284-3513. Ed Beck.

For Sale: Near new TRS-80 microcomputer. First cash offer of $600. Contact Don Christen, Executive Vice President, Contra Costa Taxpayers Assn., Martinez, CA 94553. 415-228-5610.

For Sale: Xitan 16K dyn. RAM bd. Tested: Perfect on Z-80 system at 4 mHz. S-100 bus. $130 incl. post. M. Miller, 2408 NW 110th St., OKC, OK 73120, 405-755-6494.

Centronics 306C printer: 80/132 column selectable forms option, sound cover, lowercase character set, serial interface, 120 characters/sec.; $1200 or offer. Write: Michael Sloat, PO Box 982, Loma Linda, CA 92354. 714-796-2757.

Centronics 306C printer: 80/132 column selectable forms option, sound cover, lowercase character set, parallel interface for TRS-80, 120 characters/sec., $1200 or offer. Write: Michael Sloat, PO Box 982, Loma Linda, CA 92354. 714-796-2757.

Multiterm printer model T-4000: 55 character/sec., Diablo Hytype mechanism, tractor feed, ribbons, print wheels, many additional features, auto-underline, etc. $1700 or offer. Write Michael Sloat, PO Box 982, Loma Linda, CA 92354. 714-796-2757.

Diablo Hytype I Model 1200. Letter quality "Daisy Wheel" printers. Brand new units w/platen and print wheel. EIA RS-232C/CCITT interface kit, $75. Power supplies, $100. New printer, $650. There is no better buy anywhere. For Apple, TRS-80 & CP/M systems, interface info available. After 7PM. Scott Priester, 211 White Water Ct., Greer, SC 29651. 803-268-0678.

For Sale: Digital Group Computer, 48K RAM, 4 Phidecks, keyboard, software. All in dress cabinets, excellent condition, up and running. Paul Bachman, 605 Cynthia Dr., Fayette, OH 43521. 419-239-2972.

For Sale: TRS-80, 16K LII. Year old, excellent cond. 25% off/make offer. Write or call eve. Walter Sklannik, 222 W. Price St., Linden, NJ 07036. 201-862-4491.

SWTPC 6800; 12K, but I have boards for 24K just add TMS4044 ($4.25 ea.). MP-A2, AC-30, GT-6144, MP-N RPN calculator interface. CT-64 terminal 2K memory, CT-EA screen read. Free software; 2.3 BASIC, some TSC software. All manuals and schematics included. Asking $750 for everything. Victor J. Silva, 43 Ayers St., Waterbury, CT 06706. 203-757-1402, after 5 PM.

For Sale: Heath WH-89 with H-17 disks (3 drives), H-88-2 serial I/O, Z89-6 mod. kit, 48K RAM, HDOS version 1.6, Microsoft BASIC, all bus. software including AR/AP, Ledger, Payroll, etc. All works perfectly. First $2200 takes it and I'll ship. Also, Decision Data 6540 serial printer with stand, $1500 takes it. Less than 100 hrs. on all. Includes all cables, manuals and disks. Save $100 on whole system. Call Tom at 904-755-3599 after 6PM.

Sale: Centronics 730 printer, same as TRS-80 Lineprinter 2, almost new, mint condition, w/cable, doc., $560. G. Kish, 227 W. Meadow Rd., Rockaway, NJ 07866. 201-627-0052, evenings.

For Sale: SWTPC 6800, MP-A2, 20K, MP-S, CT-64, CT-VM, CIS-30+, CSS BASIC, good condition, $1100. Jim Benson, 701 Harold, Salina, KS 67401. 913-825-0946.

For Sale: Ohio Scientific C1P with 4K RAM. Original carton. Excellent condition. $295. James Ma, PO Box 508, New York, NY 10003.

Find it here!

MICRO QUIZ

(from page 26)

Answer: 88

I	S	C	A	B
1	1	2	1	2
2	2	3	2	3
3	4	5	3	5
4	7	8	5	8
5	12	13	8	13
6	20	21	13	21
7	33	34	21	34
8	54	55	34	55
9	88	89	55	89

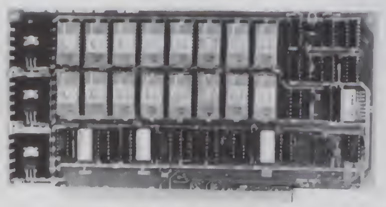
Playfair

The key to the Challenge Cipher on page 77 is EDGAR ALLEN POE.

✓Reader Service—see page 210

DIGITAL RESEARCH COMPUTERS
(214) 271-3538

32K S-100 EPROM CARD
NEW!



$74.95 KIT

USES 2716's

Blank PC Board - $34

ASSEMBLED & TESTED ADD $30

SPECIAL: 2716 EPROM's (450 NS) Are $11.95 EA. With Above Kit.

KIT FEATURES:
1. Uses +5V only 2716 (2Kx8) EPROM's.
2. Allows up to 32K of software on line!
3. IEEE S-100 Compatible.
4. Addressable as two independent 16K blocks.
5. Cromemco extended or Northstar bank select.
6. On board wait state circuitry if needed.
7. Any or all EPROM locations can be disabled.
8. Double sided PC board, solder-masked, silk-screened.
9. Gold plated contact fingers.
10. Unselected EPROM's automatically powered down for low power.
11. Fully buffered and bypassed.
12. Easy and quick to assemble.

32K SS-50 RAM

$379^{00} KIT

For 2MHZ Add $10

Blank PC Board $50

For SWTPC 6800 - 6809 Buss

Support IC's and Caps $19.95

Complete Socket Set $21.00

Fully Assembled, Tested, Burned In Add $30

NEW!

At Last! An affordable 32K Static RAM with full 6809 Capability.

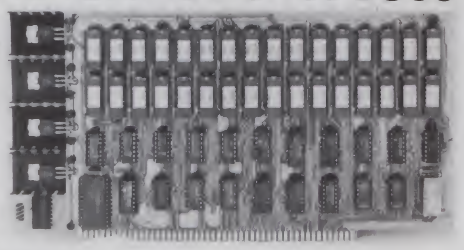
FEATURES:
1. Uses proven low power 2114 Static RAMS.
2. Supports SS50C - EXTENDED ADDRESSING.
3. All parts and sockets included.
4. Dip Switch address select as a 32K block.
5. Extended addressing can be disabled.
6. Works with all existing 6800 SS50 systems.
7. Fully bypassed. PC Board is double sided, plated thru, with silk screen.

16K STATIC RAM KIT-S 100 BUSS

PRICE CUT!

$199^{95} KIT

FOR 4MHZ ADD $10



KIT FEATURES:
1. Addressable as four separate 4K Blocks.
2. ON BOARD BANK SELECT circuitry (Cromemco Standard!). Allows up to 512K on line!
3. Uses 2114 (450NS) 4K Static Rams.
4. ON BOARD SELECTABLE WAIT STATES.
5. Double sided PC Board, with solder mask and silk screened layout. Gold plated contact fingers.
6. All address and data lines fully buffered.
7. Kit includes ALL parts and sockets.
8. PHANTOM is jumpered to PIN 67.
9. LOW POWER: under 1.5 amps TYPICAL from the +8 Volt Buss.
10. Blank PC Board can be populated as any multiple of 4K.

BLANK PC BOARD W/DATA-**$33**

LOW PROFILE SOCKET SET-**$12**

SUPPORT IC'S & CAPS-**$19.95**

ASSEMBLED & TESTED-**ADD $35**

OUR #1 SELLING RAM BOARD!

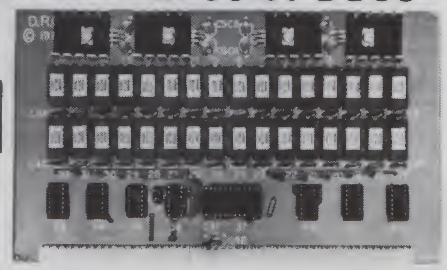
16K STATIC RAM SS-50 BUSS

PRICE CUT!

$195 KIT

FULLY STATIC!

FOR 2MHZ ADD $10



FOR SWTPC 6800 BUSS!

ASSEMBLED AND TESTED - $35

KIT FEATURES
1. Addressable on 16K Boundaries
2. Uses 2114 Static Ram
3. Fully Bypassed
4. Double sided PC Board. Solder mask and silk screened layout
5. All Parts and Sockets included
6. Low Power: Under 1.5 Amps Typical

BLANK PC BOARD—**$35** COMPLETE SOCKET SET—**$12**

SUPPORT IC'S AND CAPS—**$19.95**

STEREO! S-100 SOUND COMPUTER BOARD
NEW! NEW!

At last, an S-100 Board that unleashes the full power of two unbelievable General Instruments AY3-8910 NMOS computer sound IC's. Allows you under total computer control to generate an infinite number of special sound effects for games or any other program. Sounds can be called in BASIC, ASSEMBLY LANGUAGE, etc.

COMPLETE KIT! $84^{95} (WITH DATA MANUAL)

BLANK PC BOARD W/DATA $31

KIT FEATURES:
- TWO GI SOUND COMPUTER IC'S.
- FOUR PARALLEL I/O PORTS ON BOARD.
- USES ON BOARD AUDIO AMPS OR YOUR STEREO.
- ON BOARD PROTO TYPING AREA.
- ALL SOCKETS, PARTS AND HARDWARE ARE INCLUDED.
- PC BOARD IS SOLDERMASKED, SILK SCREENED, WITH GOLD CONTACTS.
- EASY, QUICK, AND FUN TO BUILD, WITH FULL INSTRUCTIONS.
- USES PROGRAMMED I/O FOR MAXIMUM SYSTEM FLEXIBILITY.

Both Basic and Assembly Language Programming examples are included.

SOFTWARE:

SCL™ is now available! Our Sound Command Language makes writing Sound Effects programs a SNAP! SCL™ also includes routines for Register-Examine-Modify, Memory-Examine-Modify, and Play-Memory. SCL™ is available on CP/M* compatible diskette or 2708 or 2716. Diskette - **$24.95** 2708 - **$19.95** 2716 - **$29.95**. Diskette includes the source. EPROM'S are ORG at E000H.

4K DYNAMIC RAM BLOWOUT!
SAME AS INTEL 2107B!
4K RAMS AT AN UNBELIEVABLE 50¢ EACH!!!

Prime, new, National Semi., 1979 date coded, full spec. parts. N.S. #MM5280-5N. Same as INTEL 2107B-4, T.I. TMS4060, NEC uPD411, etc. We bought a HUGE QTY. from a West Coast Distributor at truly DISTRESS PRICES! One of the most popular and reliable RAM's ever made. These parts have been used by almost all Major Computer Main Frame Mfg. the world over! Arranged as 4K x 1, 270 NS Access Time, 22 Pin Dip. These units DO NOT use multiplexed addressing, thus making REFRESH and other timing very simple. See INTEL MEMORY DESIGN HANDBOOK for full application notes. The NAT. SEMI. MEMORY DATA BOOK is available at most Radio Shack Stores. Prime units in original factory tubes!

#5280-5N 4096 BITS x 1 270 NS ACCESS

(With Pin Out Data)

8 FOR $4.95 32 FOR $16

FACTORY CASE (450 PCS) — $180

Sockets Special: 22 Pin Low Profile (With Purchase of 5280's) 8 FOR $1.

COMPUTER PARTS SPECIALS

74LS175 - .99	8035 Intel Single Chip CPU 6.95
74LS240 - 1.19	Signetics 2901 4 Bit Slice - 6.95
74LS241 - 1.19	AMD 2903 4 Bit Super Slice - 12.50
74LS244 - 1.19	AMD 29705 Dual Port RAM - 8.95
74LS373 - 1.29	Intel 2716-1 (350 NS) - 12.95

NEW! G.I. COMPUTER SOUND CHIP

AY3-8910. As featured in July, 1979 BYTE! A fantastically powerful Sound & Music Generator. Perfect for use with any 8 Bit Microprocessor. Contains: 3 Tone Channels, Noise Generator, 3 Channels of Amplitude Control, 16 bit Envelope Period Control, 2-8 Bit Parallel I/O, 3 D to A Converters, plus much more! All in one 40 Pin DIP. Super easy interface to the S-100 or other busses. $11.95 PRICE CUT!

SPECIAL OFFER: ~~$14.95~~ each Add $3 for 60 page Data Manual.

Digital Research Computers
(OF TEXAS)

P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Add $1.50 postage. We pay balance. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tex. Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. Orders over $50, add 85¢ for insurance.

ALL SALES ARE MADE SUBJECT TO THE TERMS OF OUR 90 DAY LIMITED WARRANTY. A COPY OF THIS WARRANTY IS AVAILABLE FREE, ON REQUEST.

*TRADEMARK OF DIGITAL RESEARCH.

WE ARE NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE SUPPLIERS OF CPM SOFTWARE.

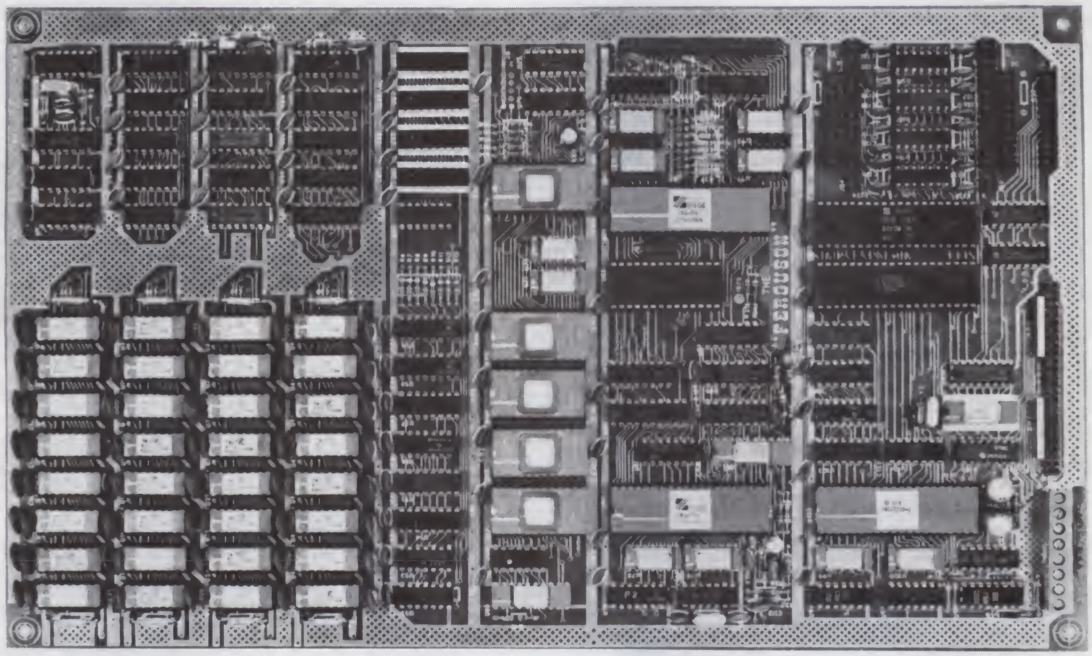
NEW! "THE BIG BOARD" NEW!

OEM - INDUSTRIAL - BUSINESS - SCIENTIFIC

SINGLE BOARD COMPUTER KIT!

Z-80 CPU! 64K RAM!

PARTIAL KIT
For All Sockets Installed
And Soldered Add $50.



U.K. AND EUROPE: CONTACT MACLIN-ZAND LTD. 38 MOUNT PLEASANT — LONDON WCIX OAP

THE FERGUSON PROJECT: Three years in the works, and maybe too good to be true. A tribute to hard headed, no compromise, high performance, American engineering! The Big Board gives you all the most needed computing features on one board at a very reasonable cost. The Big Board was designed from scratch to run the latest version of CP/M*. Just imagine all the off-the-shelf software that can be run on the Big Board without any modifications needed! Take a Big Board, add a couple of 8 inch disc drives, power supply, an enclosure, C.R.T., and you have a total Business System for about 1/3 the cost you might expect to pay.

$649.00** (64K KIT BASIC I/O)

SIZE: 8½ x 13¾ IN.
SAME AS AN 8 IN. DRIVE.
REQUIRES: +5V @ 3 AMPS
+ - 12V @ .5 AMPS.

FULLY SOCKETED!

FEATURES: (Remember, all this on one board!)

64K RAM

Uses industry standard 4116 RAM'S. All 64K is available to the user, our VIDEO and EPROM sections do not make holes in system RAM. Also, very special care was taken in the RAM array PC layout to eliminate potential noise and glitches.

Z-80 CPU

Running at 2.5 MHZ. Handles all 4116 RAM refresh and supports Mode 2 INTERUPTS. Fully buffered and runs 8080 software.

SERIAL I/O (OPTIONAL)

Full 2 channels using the Z80 SIO and the SMC 8116 Baud Rate Generator. FULL RS232! For synchronous or asynchronous communication. In synchronous mode, the clocks can be transmitted or received by a modem. Both channels can be set up for either data-communication or data-terminals. Supports mode 2 Int. Price for all parts and connectors: $85.

BASIC I/O

Consists of a separate parallel port (Z80 PIO) for use with an ASCII encoded keyboard for input. Output would be on the 80 x 24 Video Display.

24 x 80 CHARACTER VIDEO

With a crisp, flicker-free display that looks extremely sharp even on small monitors. Hardware scroll and full cursor control. Composite video or split video and sync. Character set is supplied on a 2716 style ROM, making customized fonts easy. Sync pulses can be any desired length or polarity. Video may be inverted or true. 5 x 7 Matrix - Upper & Lower Case

FLOPPY DISC CONTROLLER

Uses WD1771 controller chip with a TTL Data Separator for enhanced reliability. IBM 3740 compatible. Supports up to four 8 inch disc drives. Directly compatible with standard Shugart drives such as the SA800 or SA801. Drives can be configured for remote AC off-on. Runs CP/M* 2.2.

TWO PORT PARALLEL I/O (OPTIONAL)

Uses Z-80 PIO. Full 16 bits, fully buffered, bi-directional. User selectable hand shake polarity. Set of all parts and connectors for parallel I/O: $29.95

REAL TIME CLOCK (OPTIONAL)

Uses Z-80 CTC. Can be configured as a Counter on Real Time Clock. Set of all parts: $14.95

SYSTEM COMPARISON

64K RAM KIT	$370.00
80 x 24 Video Kit	365.00
Floppy Disk Controller Kit	235.00
Z-80 CPU Kit	185.95
SER & PAR. I/O	129.95
S-100 Mother Board	45.00
SUB TOTAL	$1330.90

Talk about bangs per buck! The prices shown for S100 kits were taken from the July 1980 BYTE. This will give some basis for comparison between the Big Board and a similar system implementation on the S100 Buss.

CP/M* 2.2 FOR BIG BOARD

The popular CP/M* D.O.S. modified by MICRONIX SYSTEMS to run on Big Board is available for $150.00.

PC BOARD

Blank PC Board with Rom Set and Full Documentation. $195.00

PFM 3.0 2K SYSTEM MONITOR

The real power of the Big Board lies in its PFM 3.0 on board monitor. PFM commands include: Dump Memory, Boot CP/M*, Copy, Examine, Fill Memory, Test Memory, Go To, Read and Write I/O Ports, Disc Read (Drive, Track, Sector), and Search. PFM occupies one of the four 2716 EPROM locations provided.

Z-80 is a Trademark of Zilog.

Digital Research Computers
(OF TEXAS)

P.O. BOX 401565 • GARLAND, TEXAS 75040 • (214) 271-3538

TERMS: Initial shipments will be made approximately 3 to 5 weeks after we receive your order. VISA, MC, cash accepted. We will accept COD's (for the Big Board only) with a $75 deposit. Balance UPS COD. The $75 deposit assures your place in line for the initial production run of Big Board.

*TRADEMARK OF DIGITAL RESEARCH. NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE ORIGINATORS OF CPM SOFTWARE
**1 TO 4 PIECE DOMESTIC USA PRICE.

WAMECO

THE COMPLETE PC BOARD HOUSE
EVERYTHING FOR THE S-100 BUSS

NEW CPU-2, Z80 PROCESSOR.

* ON BOARD ROM. (2716 OR 2732)
* HARDWARE POWER ON JUMP TO THE NEAREST 256 BYTE.
* MAJOR EMPHASIS ON PROPER TIMING.
* WILL WORK ON AN EXTENDER BOARD AT 4 MHZ.
* SWITCH SELECTABLE FOR 2 OR 4 MHZ OPERATION.
* FRONT PANEL OPERATION.

FUTURE PRODUCTS: 80 CHARACTER VIDEO BOARD.
Z-80 CPU BOARD WITH ROM, 8 PARALLEL PORT I/O BOARD.

DEALER INQUIRIES INVITED, UNIVERSITY DISCOUNTS AVAILABLE AT YOUR LOCAL DEALER

WMC inc.

WAMECO. INC., P. O. BOX 877 • 455 PLAZA ALHAMBRA • EL GRANADA, CA 94018 • (415) 728-9114

CALIFORNIA COMPUTER SYSTEMS

32K RAM BOARD A&T.
450 NSEC .... $579.95, 200 NSEC $629.95
16K RAM A&T.
450 NSEC .... $255.95, 200 NSEC $285.95
64K DYNAMIC A&T.
200 NSEC $548.95
Z80 PROCESSOR A&T. $259.00
DISC CONTROLLER $339.95
APPLE IEEE INSTRUMENTATION INTERFACE KIT 7490. A & T $275.00
ARITHMETIC PROCESSOR FOR APPLE 7811A. A & T $342.80
APPLE ASYNCHRONOUS SERIAL INTERFACE 7710A. A & T $137.10
APPLE SYNCHRONOUS SERIAL INTERFACE 7712A. A & T $153.95
ALL OTHER CCS PRODUCTS AVAILABLE

SSM

PB-1 2708 & 2716 Programming Board with provisions for 4K or 8K EPROM. *No external supplies required.* Textool sockets. Kit $143.00
CB-1A 8080 Processor Board. 2K of PROM 256 BYTE RAM power on/rest Vector Jump Parallel port with status. Kit $146.00 PCBD $31.95
VB-3 80x24 VIDEO BOARD. Graphics included. 4MHZ. Kit $379.95
IO-4 Two serial I/O ports with full handshaking 20/60 ma current loop: Two parallel I/O ports. Kit .... $168.00 PCBD $31.95
VB-IC 64 x 16 video board, upper lower case Greek composite and parallel video with software, S-100. Kit $143.00
CB-2 Z80 CPU BOARD. Kit $199.95
AIO APPLE SERIAL/PARALLEL Kit $144.95
ALL OTHER SSM PRODUCTS AVAILABLE

WMC inc. WAMECO INC.

MEM-3 32K STATIC RAM 2114 24 bit addressing $36.95
FDC-1 FLOPPY CONTROLLER BOARD will drive shugart, pertek, remic 5" & 8" drives up to 8 drives, on board PROM with power boot up, will operate with CPM™ (not included). PCBD $43.95
FPB-1 Front Panel. IMSAI size, hex displays. Byte, or instruction single step. PCBD $48.50
QM-12 MOTHER BOARD, 13 slot, terminated, S-100 board only $39.95
CPU-1 8080A Processor board S-100 with 8 level vector interrupt. PCBD $28.95
RTC-1 Realtime clock board. Two independent interrupts. Software programmable. PCBD $25.95
EPM-2 2708/2716 16K/32K EPROM CARD. PCBD $28.95
QM-9 MOTHER BOARD. Short Version of QM-12. 9 Slots. PCBD $33.95
MEM-1 8K x 8 RAM 2102. Buffered PCBD $31.95
PTB-1 POWER SUPPLY AND TERMINATOR BOARD. PCBD $28.95
IOB-1 SERIAL AND PARALLEL INTERFACE. 2 parallel, one serial and cassette. PCBD $28.95

2708	$7.50	2114L 450 NSEC	$4.99
2716	$25.95	2114L 200 NSEC	$5.99

APRIL SPECIAL SALE ON PREPAID ORDERS
(Charge cards not included on this offer)

WAMECO CPU-2. PCBD $29.95
Z80 PROCESSOR WITH MIKOS PARTS $119.95

WAMECO PCBD
IOB-1, CPU-1 .... $24.95 EA.
PTB-1, RTC-1 .... $22.95 EA.

MIKOS PARTS ASSORTMENT WITH WAMECO AND CYBERCOM PCBDS

MEM-3 less RAM .... $ 95.95
With 2114L-4 .... $350.00
With 2114L-2 .... $390.00
CPU-1 with MIKOS #2 8080A CPU $99.95
QM-12 with MIKOS #4 13 slot mother board $110.95
RTC-1 with MIKOS #5 real time clock $65.95
EPM-2 with MIKOS #11 16-32K EPROMS less EPROMS $65.95
QM-9 with MIKOS #12 9 slot mother board $99.95
FPB-1 with MIKOS #14 all parts for front panel $144.95

MIKOS PARTS ASSORTMENTS ARE ALL FACTORY MARKED PARTS. KITS INCLUDE ALL PARTS LISTED AS REQUIRED FOR THE COMPLETE KIT LESS PARTS LISTED ALL SOCKETS INCLUDED

LARGE SELECTION OF LS TTL AVAILABLE
PURCHASE $50.00 WORTH OF LS TTL AND GET 10% CREDIT TOWARD ADDITIONAL PURCHASES.
PREPAID ORDERS ONLY

MIKOS

(415) 728-9121
P. O. Box 955 • El Granada, CA 94018
Please send for IC, Xistor and Computer parts list

VISA or MASTERCHARGE. Send account number, interbank number, expiration date and sign your order. Approx. postage will be added. Check or money order will be sent post paid in U.S. If you are not a regular customer, please use charge, cashier's check or postal money order. Otherwise there will be a two-week delay for checks to clear. Calif. residents add 6% tax. Money back 30-day guarantee. We cannot accept returned IC's that have been soldered to. Prices subject to change without notice. **$10 minimum order. $1.50 service charge on orders less than $10.00.**

SALE SALE SALE

Disk Drives

JADE's new dual disk sub-assemblies include: Handsome metal cabinet with proportionally balanced air flow system, rugged dual drive power supply, cooling fan, cable kit, lighted power switch, approved fuse assembly, line cord, Never-Mar rubber feet, and all necessary hardware to mount 2-8" disk drives - it's all American made, guaranteed for six months, and it's in stock!

Dual 8" Sub-Assembly Cabinet

END-000421 *Cabinet kit* **$225.00**
END-000420 *Bare cabinet* **$59.95**

Single sided, double density disk drive sub-system

END-000423 *Kit w/2 8" drives* **$975.00**
END-000424 *A & T w/2 8" drives* **$1195.00**

Double sided, double density disk drive sub-system

END-000426 *kit w/2 8" drives* **$1495.00**
END-000427 *A & T w/2 8" drives* **$1695.00**

JADE DISK PACKAGE

Double density controller, two 8" double density floppy disk drives, CP/M 2.2 (configured for controller), hardware and software manuals, boot PROM, cabinet, power supply, fan, & cables

Special package price **$1395.00**

8" Disk Drive Sale

Highly reliable double density floppy disk drives

Shugart 801R single sided, double density

MSF-10801R *SA-801R* **$425.00**
Special Sale Price .... ***2 for $790.00***

Siemens FDD100-8D2 single sided, double density

MSF-201120 *6 mo warranty* **$385.00**
Special sale price .... ***2 for $750.00***

Real Double-Sided Drives

8" Double-Sided Double-Density Sale

****

* ***Shugart SA-851R double-sided, double-density*** *

* *only* **$625.00** *ea* .... ***2 for $1190.00*** *

****

MFE M701 8" double-sided, double-density drives

only **$525** *ea* .... ***2 for $1040.00***

Qume Data Track 8 double-sided, double-density drives

only **$575.00** .... ***2 for $1100.00***

Printers

CENTRONICS 737-1

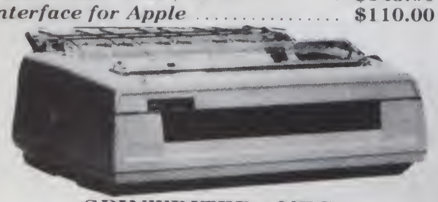
9 x N dot matrix, letter quality, proportional spacing

PRM-15737 *Parallel* **$795.00**
With interface for Apple **$895.00**

MX-80 - Epson

132 column, 9 x 9 dot matrix, multiple fonts

PRM-27080 *Save $100.00* **$545.00**
Interface for Apple **$110.00**



SPINWRITER - NEC

65 cps, bi-directional, letter quality printer with deluxe tractor mechanism, both parallel and serial interfaces on-board, 16K buffer, ribbon, print thimble, graphics, micro space justification, data cable, and self test/diagnostic ROM.

PRD-55511 *without 16K buffer* ... **$2795.00**
PRD-55512 *with 16K buffer* **$2895.00**

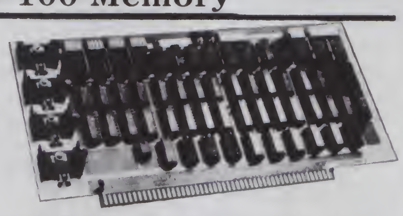
S-100 Systems

S-100 SYSTEM - Calif Computer Sys

Complete S-100 system including 12 slot mainframe, 4 MHz Z-80 CPU, 64K RAM memory, double density disk controller, RS-232 cable, 8" & 5¼" disk drive cables, CP/M 2.2, manuals, auto boot ROM, completely assembled & tested.

2210A *Integrated & tested* **$1995.00**
2210B *Not integrated* **$1795.00**

S-100 Memory



64K RAM - Calif Computer Sys

4 MHz bank port / bank byte selectable, extended addressing, 16K bank selectable, PHANTOM line allows memory overlay, 8080 / Z-80 / front panel compatible.

MEM-64565A *A & T* **$449.95**

MEMORY BANK - Jade

4 MHz, IEEE S-100, bank selectable, 8 or 16 bit

MEM-99730B *Bare board* **$55.00**
MEM-99730K *Kit, no RAM* **$219.95**
MEM-16730K *16K kit* **$249.95**
MEM-32731K *32K kit* **$289.95**
MEM-48732K *48K kit* **$324.95**
MEM-64733K *64K kit* **$359.95**
Assembled & tested .... add **$50.00**

EXPANDORAM II - S D Systems

4 MHz RAM board expandable from 16K to 256K

MEM-16630K *16K kit* **$275.95**
MEM-32631K *32K kit* **$295.95**
MEM-48632K *48K kit* **$315.95**
MEM-64633K *64K kit* **$335.95**
Assembled & tested .... add **$50.00**

32K STATIC RAM - Jade

2 or 4 MHz expandable static RAM board uses 2114L's

MEM-16151K *16K 4 MHz kit* **$169.95**
MEM-32151K *32K 4 MHz kit* **$299.95**
Assembled & tested .... add **$50.00**

16K STATIC RAM - Cal Comp Sys

2 or 4 MHz 16K static RAM board, IEEE S-100, bank selectable, Phantom capability, addressable in 4K blocks

MEM-16160A *16K 2 MHz A & T* ... **$286.95**
MEM-16162A *16K 4 MHz A & T* ... **$289.95**
MEM-16160B *Bare board* **$50.00**

PB-1 - S.S.M.

2708, 2716 EPROM board with built-in programmer

MEM-99510K *Kit* **$154.95**
MEM-99510A *A & T* **$229.95**

PROM-100 - SD Systems

2708, 2716, 2732, 2758, & 2516 EPROM programmer

MEM-99520K *Kit* **$219.95**
MEM-99520A *Jade A & T* **$269.95**

S-100 Video

VB-3 - S.S.M.

80 characters x 24 lines expandable to 80 x 48 for a full page of text, upper & lower case, 256 user defined symbols, 160 x 192 graphics matrix, memory mapped, has key board input.

IOV-1095K *4 MHz kit* **$375.00**
IOV-1095A *4 MHz A & T* **$450.00**
IOV-1096K *80 x 48 upgrade* **$39.95**

VIDEO BOARD - Jade

64 characters x 16 lines, 7 x 9 dot matrix, full upper/lower case ASCII character set, numbers, symbols, and greek letters, normal/reverse/blinking video, S-100.

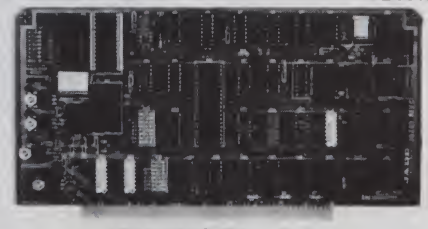
IOV-1050K *Kit* **$99.95**
IOV-1050A *A & T* **$125.00**
IOV-1050B *Bare board* **$19.95**

S-100 CPU

2810 Z-80* CPU - Cal Comp Sys

2/4 MHz Z-80A CPU with RS-232C serial I/O port and on-board MOSS 2.2 monitor PROM, front panel compatible.*

CPU-30400A *A & T* **$269.95**



THE BIG Z* - Jade

2 or 4 MHz switchable Z-80 CPU with serial I/O, accomodates 2708, 2716, or 2732 EPROM, baud rates from 75 to 9600*

CPU-30201K *Kit* **$145.00**
CPU-30201A *A & T* **$199.00**
CPU-30200B *Bare board* **$35.00**

CB-2 Z-80 CPU - S.S.M.

2 or 4 MHz Z-80 CPU board with provision for up to 8K of ROM or 4K of RAM on board, extended addressing, IEEE S-100, front panel compatible.

CPU-30300A *A & T* **$229.95**

SBC-200 - SD Systems

4 MHz Z-80 CPU with serial & parallel I/O ports, up to 8K of on-board PROM, software programmable baud rate generator, 1K of on-board RAM, Z-80 CTC.*

CPC-30200K *Kit* **$339.95**
CPC-30200A *Jade A & T* **$399.95**

S-100 Disk Controller

DOUBLE DENSITY - Cal Comp Sys

5¼" and 8" disk controller, single or double density, with on-board boot loader ROM, and free CP/M 2.2 and manual set.*

IOD-1300A *A & T* **$369.95**

DOUBLE-D - Jade

Double density controller with the inside track, on-board Z-80A, printer port, IEEE S-100, can function on an interrupt driven buss*

IOD-1200K *Kit* **$299.95**
IOD-1200A *8" A & T* **$389.95**
IOD-1205A *5¼" A & T* **$389.95**
IOD-1200B *Bare board* **$65.00**

VERSAFLOPPY II - SD Systems

New double density controller for both 8" & 5¼"

IOD-1160K *Kit* **$379.95**
IOD-1160A *Jade A & T* **$439.95**

Motherboards

ISO-BUS - Jade

Silent, simple, and on sale - a better motherboard

6 Slot (5¼" x 8⅝")

MBS-061B *Bare board* **$19.95**
MBS-061K *Kit* **$39.95**
MBS-061A *A & T* **$49.95**

12 Slot (9¾" x 8⅝")

MBS-121B *Bare board* **$29.95**
MBS-121K *Kit* **$69.95**
MBS-121A *A & T* **$89.95**

18 Slot (14½" x 8⅝")

MBS-181B *Bare board* **$49.95**
MBS-181K *Kit* **$99.95**
MBS-181A *A & T* **$139.95**

Card Cages

S-100 CARD CAGE - Jade

Metal cage with card guides & fan mounting

ENX-106001 *Six slot* **$29.95**

S-100 CARD CAGE - Vector

19" rack mountable, adjustable, holds 21 cards

VCT-CCK100 *Anodized Al* **$49.95**

SALE SALE SALE

S-100 I/O

S.P.I.C. - Jade
Our new I/O card with 2 SIO's, 4 CTC's, and 1 PIO

IOI-1045K *2 CTC's, 1 SIO, 1 PIO* .. $199.00
IOI-1045A *A & T* $259.00
IOI-1046K *4 CTC's, 2 SIO's, 1 PIO* $259.00
IOI-1046A *A & T* $319.00
IOI-1045B *Bare board w/ manual* ... $59.95
IOI-1045D *Manual only* $20.00

I/O-4 - S.S.M.
2 serial I/O ports plus 2 parallel I/O ports

IOI-1010K *Kit* $179.95
IOI-1010A *A & T* $249.95
IOI-1010B *Bare board* $35.00

TERMINATOR - S.S.M.
Active terminator for S-100 bus

TSX-195K *Kit* $29.95
TSX-195A *A & T* $54.95
TSX-195B *Bare board* $22.95

S-100 EXTENDER - Cal Comp Sys
Put those problem boards (the ones you probably bought from one of our competitors) within easy reach.

TSX-160A *A & T* $37.95

S-100 PROTO BOARD - Jade
Universal design, plated thru holes, gold fingers

TSX-140B *Bare board* $24.95

TERMINATOR & EXTENDER - C.C.S.
Can be used as both an S-100 extender and terminator

TSX-150K *Kit* $43.95

Diskettes

DISKETTES - Jade
Bargain prices on magnificent magnetic media

5¼" single sided, single density, box of 10

MMD-5110103 *Soft sector* $27.95
MMD-5111003 *10 sector* $27.95
MMD-5111603 *16 sector* $27.95

5¼" double sided, double density, box of 10

MMD-5220103 *Soft sector* $39.95

8" single sided, single density, box of 10

MMD-8110103 *Soft sector* $33.95

8" single sided, double density, box of 10

MMD-8120103 *Soft sector* $39.95

8" double sided, double density, box of 10

MMD-8220103 *Soft sector* $49.95

Video Monitors

9" B & W MONITOR - A.P.F.
High quality, high resolution video monitor

VDM-750900 *9" monitor* $159.95

13" COLOR MONITOR - Zenith
The hi res color you've been promising yourself

VDC-201301 .... $449.00

12" GREEN SCREEN - NEC
20 MHz, P31 phosphor video monitor with audio, exceptionally high resolution - A fantastic monitor at a very reasonable price

VDM-651200 *12" monitor* $259.95

Mainframes

MAINFRAME - Cal Comp Sys
12 slot S-100 mainframe with 20 amp power supply

ENC-112105 *Kit* $359.95
ENC-112106 *A & T* $419.95

DISK MAINFRAME - NNC
Holds 2 8" drives and an 8 slot S-100 system. Attractive metal cabinet with 8 slot motherboard, power supply, fan, key switch, and other professional features

ENS-112320 *with 30 amp p.s.* $699.95

Accessories-Apple/TRS-80

16K MEMORY UPGRADE
Add 16K of RAM to your TRS-80, Apple, or Exidy in just minutes. We've sold thousands of these 16K RAM upgrades which include the appropriate memory chips (as specified by the manufacturer), all necessary jumper blocks, fool-proof instructions, and our 1 year guarantee.

MEX-16100K *TRS-80 kit* $29.00
MEX-16101K *Apple kit* $29.00
MEX-16102K *Exidy kit* $29.00

DISK DRIVE for APPLE
5¼" disk drive with controller for your Apple

MSM-12310C *with controller* $499.95
MSM-123101 *w/out controller* $375.00

DISK DRIVES for TRS-80
23% more storage, 8 times faster, 40 track with free patch, 120 day warranty, includes case, power supply, and cable

MSM-12410C *Save $125.00 !!!* $299.95

DOS 3.3 UPGRADE - Apple
Upgrade your old DOS to the improved 3.3

IOD-2233A *Complete kit* $64.95

APPLE STICK - Micromate
Joy stick with pots for Apple II

SYA-1510A *A & T* $35.95

Z-80* CARD for APPLE
Z-80 CPU card with CP/M 2.2 for your Apple*

CPX-30800A *A & T* $279.95

AIO - S.S.M.
Parallel & serial interface for your Apple

IOI-2050K *Kit* $155.95
IOI-2050A *A & T* $194.95

PRINTER INTERFACE - C.C.S.
Centronics type I/O card w/ firmware

IOI-2041A *A & T* $99.95

APPLE CLOCK - Cal Comp Sys
Real time clock w/ battery back-up

IOK-2100A *A & T* $109.95

Modems

★★★★★★★★★★★★★★★★★★★★★★★★★★★★

LEX-11 MODEM - Lexicon
A real star! 300 baud, answer/originate, RS 232C

IOM-5511A *Best buy !!!* $128.00

★★★★★★★★★★★★★★★★★★★★★★★★★★★★

NOVATION CAT
300 baud, answer/originate acoustic modem

IOM-5200A *1 year warranty* $179.00

D-CAT 300 baud, direct connect modem

IOM-5201A *Special sale price* $189.00

AUTO-CAT Auto answer/origiate, direct connect

IOM-5230A *Special sale price* $239.95

MICROMODEM - D.C. Hayes
Auto answer/dial modem card for Apple or S-100

IOM-2010A *Apple modem* $349.95
IOM-1100A *S-100 modem* $375.00

MICRONET MODEM - Micromate
Direct connect with extra features - a best buy

IOM-2020A *Best Apple modem* $275.00

* Z-80, Z-80A, and the letter Z are recognized trademarks of Zilog, Inc. *CP/M is a registered trademark of Digital Research Corp. *CBASIC is a trademark of Compiler Systems, Inc.

Single Board Computers

AIM-65 - Rockwell
6502 computer with alphanumeric display, printer, & keyboard, and complete instructional manuals

CPK-50165 *1K AIM* $374.95
CPK-50465 *4K AIM* $449.95
SFK-74600008E *8K BASIC ROM* ... $99.95
SFK-64600004E *4K assembler ROM* $84.95
PSX-030A *Power supply* $64.95
ENX-000002 *Enclosure* $49.95

4K AIM, 8K BASIC, power supply, & enclosure

Special package price $625.00

Z-80* STARTER KIT - SD Systems
Complete Z-80 computer with RAM, ROM, I/O, display, keyboard, manual, and kluge area.*

CPS-30010K *Kit* $369.95
CPS-30010A *Jade A & T* $459.95

MICROPROCESSORS

Z-80	10.95
Z-80A	12.95
6502	11.50
6800	11.95
6802	17.95
6809	39.95
8035	24.00
8080A	6.59
8085	15.95
8748	59.95

Z-80 SUPPORT

3881 *PIO*	9.50
3881-4 *PIO-4 MHz*	14.50
3882 *CTC*	9.50
3882-4 *CTC-4 MHz*	14.95
3883 *SIO*	29.50
3884 *SIO*	49.50

BAUD RATE GENERATORS

MC14411	10.00
1.843 MHz xtal	4.95

UARTS

AY5-1013A	5.25
AY3-1014A	8.25
TR1602B	5.25
TMS6011	5.95
IM6402	9.00

6800 SUPPORT

6821P	5.95
6828P	11.95
6834P	22.50
6840P	18.75
6850P	4.80
6852P	5.79
6875L	7.40
68488P	25.00

PROMS

2708 *450ns*	6.25
10 for $4.90 ea	
2716 *12,5v*	11.95
2716 *5v*	11.95
10 for $8.90 ea	
2532 *5v*	39.95
2732 *5v*	39.95
2758 *5v*	9.95

RAMS

21L02 *2 MHz*	1.25
21L02A *4 MHz*	1.50
2114L *2 MHz*	3.75
2114LA *4 MHz*	3.95
4116	4.25
4164 *64K x1*	59.95
5257 *2 MHz*	6.75
5257A *4 MHz*	7.25
MK4118	18.95

SUPPORT DEVICES

8212	3.25
8214	4.65
8216	2.95
8224	3.25
8224-4	5.75
8226	3.85
8228	4.95
8238	4.95
8243	8.00
8250	14.95
8251	6.50
8253	17.95
8255	6.50
8257	19.95
8259	17.95
8275	49.95
8279	15.95

PLACE ORDERS TOLL FREE

Continental U.S. 800-421-5500
Inside California 800-262-1710

For Technical Inquiries or Customer Service call

48

213-973-7707

JADE
Computer Products

4901 W. Rosecrans, Hawthorne, Ca 90250

TERMS OF SALE: Cash, checks, credit cards, or Purchase Orders from qualified firms and institutions. Minimum order $15.00. California residents add 6% tax. Minimum shipping and handling charge $3.00 **Pricing and availability subject to change without notice.**

✓Reader Service—see page 210

✓Reader Service—see page 210

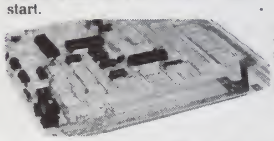
Start learning and computing for only $129.95 with a Netronics 8085-based computer kit. Then expand it in low-cost steps to a business/development system with 64k or more RAM, 8" floppy disk drives, hard disks and multi-terminal I/O.

THE NEW EXPLORER/85 SYSTEM

Special! *Full 8" floppy, 64k system for less than the price of a mini! Only* **$1499.95!**

(Also available wired & tested, $1799.95)

Imagine — for only $129.95 you can own the starting level of Explorer/85, a computer that's expandable into full business/development capabilities — a computer that can be your beginner system, an OEM controller, or an IBM-formatted 8" disk small business system. From the first day you own Explorer/85, you begin computing on a significant level, and applying principles discussed in leading computer magazines. Explorer/85 features the advanced Intel 8085 cpu, which is 100% compatible with the older 8080A. It offers onboard S-100 bus expansion, Microsoft BASIC in ROM, plus instant conversion to mass storage disk memory with standard IBM-formatted 8" disks. All for only $129.95, plus the cost of power supply, keyboard/terminal and RF modulator if you don't have them (see our remarkable prices below for these and other accessories). With a Hex Keypad/display front panel, Level "A" can be programmed with no need for a terminal, ideal for a controller, OEM, or a real low-cost start.



Level "A" is a complete operating system, perfect for beginners, hobbyists, industrial controller use. $129.95

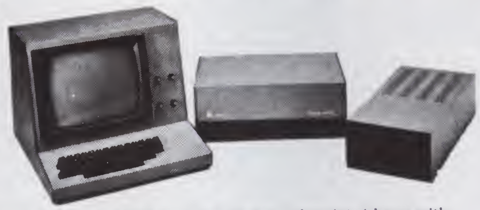
LEVEL "A" SPECIFICATIONS

Explorer/85's Level "A" system features the advanced Intel 8085 cpu, an 8355 ROM with 2k deluxe monitor/operating system, and an advanced 8155 RAM I/O . . . all on a single motherboard with room for RAM/ROM/PROM/EPROM and S-100 expansion, plus generous prototyping space.

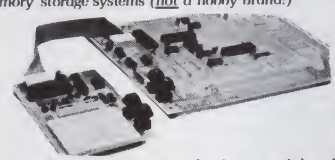
PC Board: Glass epoxy, plated through holes with solder mask. • **I/O:** Provisions for 25-pin (DB25) connector for terminal serial I/O, which can also support a paper tape reader . . . cassette tape recorder input and output . . . cassette tape control output . . . LED output indicator on SOD (serial output) line . . . printer interface (less drivers) . . . total of four 8-bit plus one 6-bit I/O ports. • **Crystal Frequency:** 6.144 MHz. • **Control Switches:** Reset and user (RST 7.5) interrupt . . . additional provisions for RST 5.5, 6.5 and TRAP interrupts onboard. • **Counter/Timer:** Programmable, 14-bit binary. • **System RAM:** 256 bytes located at F800, ideal for smaller systems and for use as an isolated stack area in expanded systems . . . RAM expandable to 64K via S-100 bus or 4k on motherboard.

System Monitor (Terminal Version): 2k bytes of deluxe system monitor ROM located at F000, leaving 0000 free for user RAM/ROM. Features include tape load with labeling . . . examine/change contents of memory . . . insert data . . . warm start . . . examine and change all registers . . . single step with register display at each break point, a debugging/training feature . . . go to execution address . . . move blocks of memory from one location to another . . . fill blocks of memory with a constant . . . display blocks of memory . . . automatic baud rate selection to 9600 baud . . . variable display line length control (1-255 characters/line) . . . channelized I/O monitor routine with 8-bit parallel output for high-speed printer . . . serial console in and console out channel so that monitor can communicate with I/O ports.

System Monitor (Hex Keypad/Display Version): Tape load with labeling . . . tape dump with labeling . . . examine/change contents of memory . . . insert data . . . warm start . . . examine and change all registers . . .



Full 8" disk system for less than the price of a mini (shown with Netronics Explorer/85 computer and new terminal). System features floppy drive from Control Data Corp., world's largest maker of memory storage systems (not a hobby brand!)



Level "A" With Hex Keypad/Display.

single step with register display at each break point . . . go to execution address. Level "A" in this version makes a perfect controller for industrial applications, and is programmed using the Netronics Hex Keypad/Display. It is low cost, perfect for beginners.

HEX KEYPAD/DISPLAY SPECIFICATIONS

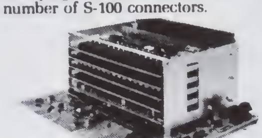
Calculator type keypad with 24 system-defined and 16 user-defined keys. Six digit calculator-type display, that displays full address plus data as well as register and status information.

LEVEL "B" SPECIFICATIONS

Level "B" provides the S-100 signals plus buffers/drivers to support up to six S-100 bus boards, and includes: address decoding for onboard 4k RAM expansion selectable in 4k blocks . . . address decoding for onboard 8k EPROM expansion selectable in 8k blocks . . . address and data bus drivers for onboard expansion . . . wait state generator (jumper selectable), to allow the use of slower memories . . . two separate 5 volt regulators.

LEVEL "C" SPECIFICATIONS

Level "C" expands Explorer/85's motherboard with a card cage, allowing you to plug up to six S-100 cards directly into the motherboard. Both cage and card are neatly contained inside Explorer's deluxe steel cabinet. Level "C" includes a sheet metal superstructure, a 5-card, gold plated S-100 extension PC board that plugs into the motherboard. Just add required number of S-100 connectors.



Explorer/85 With Level "C" Card Cage.

LEVEL "D" SPECIFICATIONS

Level "D" provides 4k of RAM, power supply regulation, filtering decoupling components and sockets to expand your Explorer/85 memory to 4k (plus the original 256 bytes located in the 8155A). The static RAM can be located anywhere from 0000 to EFFF in 4k blocks.

LEVEL "E" SPECIFICATIONS

Level "E" adds sockets for 8k of EPROM to use the popular Intel 2716 or the TI 2516. It includes all sockets, power supply regulator, heat sink, filtering and decoupling components. Sockets may also be used for 2k x 8 RAM IC's (allowing for up to 12k of onboard RAM).

DISK DRIVE SPECIFICATIONS

- 8" CONTROL DATA CORP. professional drive.
- LSI controller.
- Write protect.
- Single or double density.
- Data capacity: 401,016 bytes (SD), 802,032 bytes (DD), unformatted.
- Access time: 25ms (one track).

DISK CONTROLLER/ I/O BOARD SPECIFICATIONS

- Controls up to four 8" drives.
- 1771A LSI (SD) floppy disk controller.
- Onboard data separator (IBM compatible).
- 2 Serial I/O ports
- Autoboot to disk system when system reset.
- 2716 PROM socket included for use in custom applications.
- Onboard crystal controlled.
- Onboard I/O baud rate generators to 9600 baud.
- Double-sided PC board (glass epoxy.)

DISK DRIVE CABINET/POWER SUPPLY

- Deluxe steel cabinet with individual power supply for maximum reliability and stability.

ORDER A COORDINATED EXPLORER/85 APPLICATIONS PAK!

Beginner's Pak (Save $26.00!) — Buy Level "A" (Terminal Version) with Monitor Source Listing and AP-1 5-amp Power Supply: (regular price $199.95), now at SPECIAL PRICE: **$169.95** plus post. & insur.

Experimenter's Pak II (Save $53.40!) — Buy Level "A" (Hex Keypad/Display Version) with Hex Keypad/Display, Intel 8085 User Manual, Level "A" Hex Monitor Source Listing, and AP-1 5-amp Power Supply: (regular price $279.35), all at SPECIAL PRICE: **$219.95** plus post. & insur.

Special Microsoft BASIC Pak (Save $103.00!) — Includes Level "A" (Terminal Version), Level "B", Level "D" (4k RAM), Level "E", 8k Microsoft in ROM, Intel 8085 User Manual, Level "A" Monitor Source Listing, and AP-1 5-amp Power Supply: (regular price $439.70), now yours at SPECIAL PRICE: **$329.95** plus post. & insur.

ADD A TERMINAL WITH CABINET, GET A FREE RF MODULATOR: Save over $114 at this SPECIAL PRICE: **$499.95** plus post. & insur.

Special 8" Disk Edition Explorer/85 (Save over $104!) — Includes disk-version Level "A", Level "B", two S-100 connectors and brackets, disk controller, 64k RAM, AP-1 5-amp power supply, Explorer/85 deluxe steel cabinet, cabinet fan, 8" SD/DD disk drive from famous CONTROL DATA CORP. (not a hobby brand!), drive cabinet with power supply, and drive cable set-up for two drives. This package includes everything but terminal and printers (see coupon for them). Regular price $1630.30, all yours in kit at SPECIAL PRICE: **$1499.95** plus post. & insur. Wired and tested, only **$1799.95.**

Special! Complete Business Software Pak (Save $625.00!) — Includes CP/M 2.0, Microsoft BASIC, General Ledger, Accounts Receivable, Accounts Payable, Payroll Package: (regular price $1325), yours now at SPECIAL PRICE: **$699.95.**

Please send the items checked below:

- ☐ **Explorer/85 Level "A" kit** (Terminal Version) . . . **$129.95** plus $3 post. & insur.
- ☐ **Explorer/85 Level "A" kit** (Hex Keypad/Display Version) . . . **$129.95** plus $3 post. & insur.
- ☐ **8k Microsoft BASIC** on cassette tape, **$64.95** postpaid.
- ☐ **8k Microsoft BASIC in ROM kit** (requires Levels "B", "D" and "E") . . . **$99.95** plus $2 post. & insur.
- ☐ **Level "B" (S-100) kit** . . . **$49.95** plus $2 post. & insur.
- ☐ **Level "C" (S-100 6-card expander) kit** . . . **$39.95** plus $2 post. & insur.
- ☐ **Level "D" (4k RAM) kit** . . . **$69.95** plus $2 post. & insur.
- ☐ **Level "E" (EPROM/ROM) kit** . . . **$5.95** plus 50¢ p&h.
- ☐ **Deluxe Steel Cabinet** for Explorer/85 . . . **$49.95** plus $3 post. & insur.
  - ☐ Fan For Cabinet . . . **$15.00** plus $1.50 post. & insur.
- ☐ **ASCII Keyboard/Computer Terminal kit:** features a full 128 character set, u&l case; full cursor control; 75 ohm video output; convertible to baudot output; selectable baud rate, RS232-C or 20 ma. I/O, 32 or 64 character by 16 line formats, and can be used with either a CRT monitor or a TV set (if you have an RF modulator) . . . **$149.95** plus $3.00 post. & insur.
  - ☐ **DeLuxe Steel Cabinet** for ASCII keyboard/terminal . . . **$19.95** plus $2.50 post. & insur.
- ☐ **New! Terminal/Monitor:** (See photo) Same features as above, except 12" monitor with keyboard and terminal is in deluxe single cabinet; kit . . . **$399.95** plus $7 post. & insur.
- ☐ **Hazeltine terminals:** Our prices too low to quote — **CALL US**
- ☐ **Lear-Sigler terminals/printers:** Our prices too low to quote: **CALL US**
- ☐ **Hex Keypad/Display kit** . . . **$69.95** plus $2 post. & insur.
- ☐ **AP-1 Power Supply Kit** ±8V @ 5 amps) in deluxe steel cabinet . . . **$39.95** plus $2 post. & insur.
- ☐ **Gold Plated S-100 Bus Connectors** . . . **$4.85** each, postpaid.
- ☐ **RF Modulator kit** (allows you to use your TV set as a monitor) . . . $8.95 **postpaid.**
- ☐ **16k RAM kit** (S-100 board expands to 64k) . . . **$199.95** plus $2 post. & insur.
- ☐ **32k RAM kit** . . . **$299.95** plus $2 post. & insur.
- ☐ **48k RAM kit** . . . **$399.95** plus $2 post. & insur.
- ☐ **64k RAM kit** . . . **$499.95** plus $2 post. & insur.
- ☐ **16k RAM Expansion kit** (to expand any of the above in 16k blocks up to 64k) . . . **$99.95** plus $2 post. & insur. each.
- ☐ **Intel 8085 cpu Users' Manual** . . . **$7.50** postpaid.
- ☐ **12" Video Monitor** (10MHz bandwidth) . . . **$139.95** plus $5 post. & insur.
- ☐ **Beginner's Pak** (see above) **$169.95** plus $4 post. & insur.
- ☐ **Experimenter's Pak** (see above) . . . **$219.95** plus $6 post. & insur.
- ☐ **Special Microsoft BASIC Pak Without Terminal** (see above) . . . **$329.95** plus $7 post. & insur.
- ☐ Same as above, plus **ASCII Keyboard Terminal With Cabinet, Get Free RF Modulator** (see above) . . . **$499.95** plus $10 post. & insur.
- ☐ **Special 8" Disk Edition Explorer/85** (see above) . . . **$1499.95** plus $26 post. & insur.
  - ☐ **Wired & Tested** . . . **$1799.95** plus $26 post. & insur.
- ☐ **Extra 8" CDC Floppy Drives** . . . **$499.95** plus $12 post. & insur.
  - ☐ **Cabinet & Power Supply For Drive** . . . **$69.95** plus $3 post. & insur.
- ☐ **Drive Cable Set-up For Two Drives** . . . **$25** plus $1.50 post. & insur.
- ☐ **Disk Controller Board With I/O Ports** . . . **$199.95** plus $2 post. & insur.
- ☐ **Special: Complete Business Software Pak** (see above) . . . **$699.96** postpaid.
  - SOLD SEPARATELY:
  - ☐ **CP/M 1.4** . . . **$100** postpaid.
  - ☐ **CP/M 2.0** . . . **$150** postpaid.
  - ☐ **Microsoft BASIC** . . . **$325** postpaid.
- ☐ **Intel 8085 cpu User Manual** . . . $7.50 **postpaid.**
- ☐ **Level "A" Monitor Source Listing** . . . **$25** postpaid.

Dept. KB 4

Continental U.S.A. Credit Card Buyers Outside Connecticut

CALL TOLL FREE: 800-243-7428

To Order From Connecticut Or For Technical Assistance, call (203) 354-9375

Total Enclosed (Conn res. add sales tax) $ ________

Paid By:

☐ Personal Check ☐ Cashier's Check/Money Order

☐ **VISA** ☐ **Master Charge** (Bank No. _ _ _ _)

Acct. No. ________________ Exp. Date ______

Signature ________________________

Print Name ________________________

Address ________________________

City ________________________

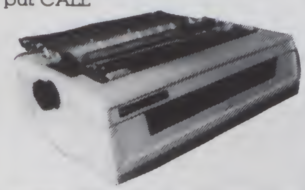
State ____________ Zip ____________

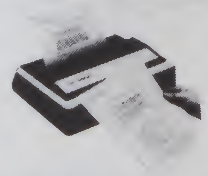
NETRONICS Research & Development Ltd.
333 Litchfield Road, New Milford, CT 06776

Reader Service—see page 210

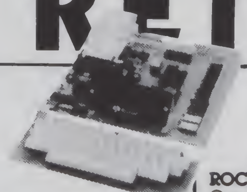
10 DAY FREE RETURN

NEC THE FIRST NAME IN LETTER QUALITY PRINTERS.
CompuMart offers beautiful print quality with NEC Spinwriter terminals. Both KSR and RO versions give unsurpassed hard copy output CALL





CENTRONICS PRINTERS
The incredible Model 737. The closest thing to letter quality print for under **$1,000.**
737-1 (Parallel Interface)—**$899.**



COMPUMART's Microflex 65 System for your AIM Includes: Adapter Buffer Module w/4-slot module stack, 8K RAM module, 16K PROM/ROM module, Asynchronous communications Interface, and Power Supply **$1,299**
Call or write for our complete Microflex 65 brochure

ROCKWELL AIM 65
Our AIM system includes: 4K AIM with BASIC interpretor assembler, Power Supply, Cassette recorder & Enclosure **$799.**
4K AIM-65 **$499.**
PL65 High Level Language **$125.**
Paper for the AIM (roll) **$2.50**
Rockwell's 4-slot Motherboard **(sale)** **$175.**
Fourth for Rockwell AIM 65 **$175.**

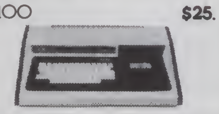
NEW FROM INTEGRAL DATA—THE IDS 560 PRINTER. 132 column graphics printer.
$1,695

IDS 445. Priced lower than the 440 and equipped with a better print head. IDS 445 w/graphics **$894.** IDS 445 w/o graphics **$795.**

IDS 460 $1,295
The 460's features include: Correspondence quality printing, high resolution graphics capability, programmable print justification.

Omni Printers from Texas Instruments
The 810—List $1895. **SALE! $1795.**
The 820 (Ro) Package—Includes machine-mounted paper tray and cable. A compressed print option and device forms control are standard features. **$2,155.**
The 820 (KSR) Package—Includes full ASCII Keyboard plus all of the features of the RO **$2395.**

NEW-IN STOCK: The PMC-80. The new computer that's software compatible with the TRS-80.
Level II 16K at **$645.**
ACCESSORIES FOR PMC—80
EXP-100 S-100 Bus Expander Disk, Printer, RS232 I/O **$410.**
S-32K S-100 Bus 32K RAM Board for EXP-100 **$295**
CAB-40 Cable 12" long ribbon cable for EXP-100 **$25.**



HP-85—Hewlett-Packard's Personal Computer for Industry. This extremely portable computer features extended BASIC to solve your problems quickly and efficiently along with an advanced graphics system to enhance communication.

HP-85 ACCESSORIES—We carry HP. Peripherals (Disk Drives to Graphics Plotters) Enhancements: (BASIC Training, General Statistics, Financial Decision, Math, Linear Programming **$95 ea.**); HP-85 Accessories: (Enhancement ROMs, ROM drawer, Overhead Transparency Kit); Supplies: (Plotter Pens, Tape Cartridges); Interface Modules (HP-IB Interface, HP-IB Interconnect Cables, Serial (RS-232C) Interface Module). We can get your every HP peripheral made for the HP-85.

VISICALC PLUS FOR HP-85 Everything you ever wanted from Visicalc plus plotting. **$200.**

HP83 VIRGO Has the same computing power as the HP85 without the added cost of tape drive and printer, for use with disc drives and optional printer/plotter accessories.
CALL FOR COMPLETE DETAILS & SPECS.

Dysan Diskettes—Single side, Single density, Hard or Soft Sector **$5. ea.**
Memorex 3401's—5 1/4 discs **$3.25** with hub ring for Apple **$3.50.**

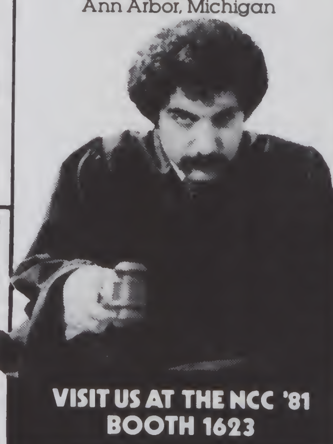
Memory Integrated Circuits—Call for qty. discounts when ordering over 50 units.
Motorola 4116 (200 Nanosecond, Plastic) **$4.50 ea.**
Fairchild 2114 (Standard Power, Plastic) **$4.50 ea.**

COMPUMART stocks the complete line of **MATROX PRODUCTS.** Call for specs.

COMPUMART now offers the ENTIRE **DEC LSI-11** PRODUCT LINE. Call for prices & delivery.

NOVATION CAT™ ACOUSTIC MODEM
Answer Originate, 300 Baud, Bell 108, Low Profile Design. **$179.00**
NEW! D-CAT
Direct Coupler from Novation **$199.**

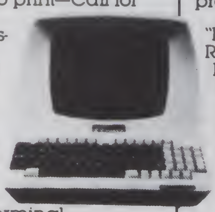
Visit our giant
ANN ARBOR STORE
1250 North Main Street
Ann Arbor, Michigan



VISIT US AT THE NCC '81 BOOTH 1623

SEND FOR OUR FREE CATALOG

Super Selling Terminals
We have the following **Lear Siegler** Terminals in stock at prices too low to print—Call for quotes.
ADM—3A Industries favorite dumb terminal for some very smart reasons.
ADM—3A + **NEW** from Lear Siegler. CALL!
IT IS HERE!—It is the **new** Intermediate Terminal from Lear Siegler. Call for details.



HP-41C Calculator **$239.00**
Memory Modules. For storing programs or up to 2,000 lines of program memory **$29.95**
"Extra Smart" Card Reader. Records programs and data back onto blank magcards. **$199.00**
The Printer. Upper and Lower case. High resolution plotting. Portable Thermal operation **$355.00**
Application Modules **$29.95**

We have the following best-selling Hazeltine terminals in stock at prices much too low to print.
(1410, 1420, 1500, 1520, 1552)
Call COMPUMART Now for our lowest prices ever.

NEW SUPER 41-C Systems with Quad RAMS built-in. (Maximum memory on-board leaves slots open for Application Pacs and peripherals. The HP 41C-C **$325.00**
HP 41C-C Plus Card Reader **$495.00**
HP 41C-C Plus Card Reader & Printer **$845.00**
Quad RAMS for the 41-C (Equivalent to four Memory Modules all packed into one) **$95.00**

EXCLUSIVE from COMPUMART! Special Offer. Zenith Color Video Monitor for $379!

NEW FROM SANYO—Four Great Monitors at Low COMPUMART Prices. Sanyo's new line of CRT data display monitors are designed for the display of alphanumeric or graphic data.
9" Sanyo Monitor **$169.**
12" Sanyo Monitor **$289.**
12" Sanyo Monitor with green screen **$299.**
13" Sanyo Color Display Monitor **$495.**

COMPUMART exclusive ATARI SPECIALS—ATARI 800 Personal Computer System—Comes with 800 operators Manual, 16K Rany Memory module, 10k ROM Operating System, Power Supply, TV Switch Box **$950.**

PERIPHERALS
Atari 410 Program Recorder **$89.95**
Atari 810 Disk Drive ($100 off with purchase) **$699.95**
NEW Dual Disk double density **$1499.95**
825 Printer (Centronics 737) **$995.00**
RS232 Interface w/Cable **$249.95**
NEW! Light Pens **$74.95**
NEW! Visicalc for Atari **$199.00**
Educators Take Note. Atari has extended its 3 for 2 deal until 6-30-81. Any educational institution that buys two Atari 800's will receive an Atari 400 computer absolutely **FREE.** Call our sales dept. for complete details.

IMPORTANT ORDERING INFORMATION All orders must include 4% **shipping and handling.** Mass. residents add 5% sales tax. Michigan residents 4% for sales tax. **Phones** open from 8:30 a.m. to 7:00 p.m., Mon.-Fri.; 11:00 a.m. to 4:00 p.m. Sat. **P.O.'s** accepted from Dun & Bradstreet rated companies—shipment contingent upon receipt of signed purchase order. **Sale prices** valid for month of magazine date only—all prices subject to change without notice. Our Ann Arbor retail store is open 11:00 a.m. to 7:00 p.m. Tues.-Fri., 10:00 a.m. to 5:00 p.m. Saturdays.

APPLE II
We carry the most complete inventory of Apple computers, peripherals, and software. CALL!
Our Best Selling Apple System: Save over $250 on our most popular Apple System. System includes a 48K Apple II, Apple Disk, DOS3.3, & Controller, and a Sup R. Mod RF Modulator.
List: **$2209.**
COMPUMART Sale Price: $1895.

SOFTWARE FROM APPLE
Apple Plot The perfect graphic complement for Visicalc. **$70**
Dow Jones News & Quotes **$95**
Adventure (Uses 48K) **$35**
DOS Tool Kit **$75**
Apple Fortran **$200**
Tax Planner **$120**

FROM PERSONAL software
Visicalc **$149**
Desk Top Plan **$99**
NEW FROM MUSE
The Voice **$39.95**
Super Text **$150**
Address Book **$49.95**

Miscellaneous Apple II Accessories:
Easy Writer (80 col. need a Videx) **$249**
Easy Mover **$49**
Easy Mailer **$69**

NEW from Apple for the Apple II.
DOS 3.3 Convert disks to 16 sector format for 23% more storage and faster access **$60.**

NEW for the Apple II
From MicroSoft
16K RAM Board **$195.00**
FORTRAN **$175.00**
COBOL **$750.00**
Card Reader from Mountain Hardware **$1,195.00**
HI-Res Dump for 460 Printer from the Computer Station **$49.95**

HARDWARE ACCESSORIES FOR APPLE
Silentype Printer w/x face **$635**
Super Sound Generator (mono) **$159** (stereo) **$259**
Light Pen **$249**
X-10 Controller (plugs into paddle port) **$49**
Mountain Computer—Expansion accessories for your Apple
Introl—10 System **$289**
Super Talker **$299**
The Music System **$545**
ROM plus board w/Keyboard filter **$199**
Clock Calendar **$280**
16 Channel A to D Converter **$350**

Apple Expansion Chassis **$650**
ROM Writer **$175**
From VIDEX!—Video Term.
80 Col. x 24 line, 7 x 9 Matrix plug in compatible board for the Apple II. Price **$325** without graphics EPROM. With graphics EPROM **$350.**
S.S.M. Serial & Parallel Apple Interface **$225**
ABT's Numeric Key Plan **$110**
California Microcomputer Keyboard **$195**
Apple III is in stock—CALL!

Commodore Strikes Back!

CompuMart carries the entire Commodore line of computers, peripherals, and accessories— call our sales force today for complete prices and specs.

The CompuMart/Commodore Word Processing System
Get crisp, letter quality output and ease of operation that's unsurpassed. This system includes a Commodore 8032 32K CPU, a 4040 Dual Disk, a C-Itoh printer and X/face, and Word Pro 4 Plus (all cables included). Call for details and low prices. List $5685. **COMPUMART $4995.**

A complete system including a Dual Disk Drive, Tractor Printer, and an 80 column 32K CPU for under $4,000. No interfaces needed. Cables included. List $3,985. **CompuMart $3635**

New accessories from Commodore for Commodore
Visicalc **$199**
Word Pro 4 **$299**
Ozz the Information Wizard **$395**
Wordcraft 80 **$395**
Tax Preparation Planner **$495**
Dow Jones Portfolio Management System **$149**
Personal Tax Calculator **$69**
PASCAL Development Package **$295**
Assembler Development Package **$99**
Educators Take Note!! Commodore has extended its 3 for 2 deal until 6-30-81.

Complete Word Processing System for Under $5,000!

COMPUMART has been serving the computer needs of industry since 1971.

We stock, for immediate shipment, only those products from the finest micro-computer manufacturers.

And any product, except software, can be returned within 10 days for a full refund—even if you just change your mind.

We also honor all manufacturers' warranties. Our expert technicians will service any product we sell that cannot be better, or faster, serviced by the manufacturer's local service center.

Call us for more information on products, product configuration and service. Our phones are open Monday thru Friday, 8:30 a.m. to 7:00 p.m. and Saturday 11:00 a.m. to 4:00 p.m.

We have a staff of highly knowledgeable sales people waiting to hear from you, and to help.

Because service is what we're all about.

VISA MasterCard

800-343-5504

In Mass. Call 617-491-2700

SYSTEMS Customized

COMPUMART ✓43
65 BENT STREET, DEPT. 135, P.O. BOX 568
CAMBRIDGE, MA 02139

Computers, Disk Systems

✓50

ZENITH data systems

Z89-FA
List $2895

OUR PRICE $2395

Z-89GA
List $2595........................ **$2149**

A-87 Two Drive
Minifloppy System List $1195........ **$989**

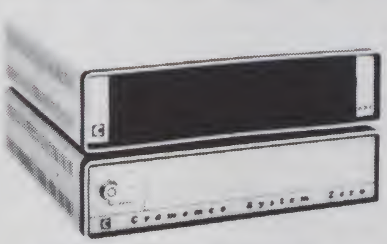
Z-47DA............................ **$3695**
8'' Two Megabyte List $3695........ **$3059**

INTERSYSTEMS
formerly ITHACA AUDIO

The new Series II CPU Board features a 4 MHz Z-80A CPU and a full-feature front panel. 20-slot actively terminated motherboard, with 25 amp power supply (50/60 Hz operation, incl. 68 cfm fan). DPS-1. List $1795

$1495

COMPLETE SYSTEM with InterSystem 64K RAM, I/O Board w/priority interrupt and double density disk controller board. Full 1-year warranty, List $3595 **$2995**



NEW! CROMEMCO SYSTEM ZERO

List $995 .... **OUR PRICE $849**

NEW! CROMEMCO SYSTEM ZERO/D

A complete 64K Computer with Double Density Disk Controller ...... List $2995

OUR PRICE $2545

Companion Disk drive for above — **Quad Density** — Total of 780 Kilobytes of storage on the two drives. ..... **List $1295**

OUR PRICE $1099

Only **$3644** for a complete 64K Disk System

SUPERBRAIN
By INTERTEC

64K Double or Quad Density units available. Uses two Z-80 CPU's. Commercial-type terminal with 12'' monitor. Dual double density minifloppies. Over 350 kilobytes of storage (twice that with quad density drives). Two serial RS232 ports, I/O ports standard. Expandable with optional S-100 S-100 interface. Comes with CP/MTM 2.2 operating system. MiniMicroMart can supply a wide range of CP/M development and application software.

w/64K Double Density, List $3495 .. **$2869**
w/64K Quad Density, List $3995..... **$3395**

HEWLETT PACKARD HP-41CV

HP-41 System I—$399.
(The HP-41CV and HP 82104A Card Reader) List $495.

HP-41 System II—$679.
(The HP-41CV and HP 82104A Card Reader, HP 82143A Printer/Plotter) List $840.

HEWLETT-PACKARD HP-85A
Desk-Top Computer

List $3250 **$2749**

MORROW THINKER TOYS® DISCUS M26™

26 megabytes of formatted storage
List $4,995 **$4,199**

THINKER TOYS® DISK SYSTEMS

Now includes CP/M® 2.2

Discus 2D, List $1199...... **$1019**

Discus 2D, dual-drive, List $1994...... **$1694**
Discus 2 + 2, Assem., List $1549....... **$1319**
Dual Discus 2 + 2, Assem., $2748...... **$2335**

All Morrow systems now include CP/M® 2.2

MORROW
Discus 2D's IN STOCK

NEW! CROMIX FROM CROMEMCO
A New UNIX Like Disk Operating System.
With true multi-user, multi-tasking capabilities

List $295 .... **OUR PRICE $249**

NEW! DOUBLE DENSITY CONTROLLER BOARD FROM CROMEMCO
With built-in diagnostics

16 FDC Controller, List $595 **OUR PRICE $505**

RADIO SHACK TRS-80™
10% OFF!

Terminals and Printers

Intertec EMULATOR

Software compatible with a Soroc IQ-120, Hazeltine 1500, ADM-3A or DEC VT-52. Features block mode transmission and printer port; 12" anti-glare screen; 18-key numeric keypad; full cursor control. List $895

OUR PRICE $749

NEW INTERTUBE III

List $995 **ONLY $749**

12" display, 24 x 80 format, 18-key numeric keypad, 128 upper/lower case ASCII characters. Reverse video, blinking, complete cursor addressing and control. Special user-defined control function keys, protected and unprotected fields. Line insert/delete and character insert/delete editing, eleven special line drawing symbols.

TELEVIDEO TVI-912C

Upper and lower case, 15 baud rates: 75 to 19,000 baud, dual intensity, 24 x 80 character display, 12 x 10 resolution. Numeric pad. Programmable reversible video, auxiliary port, self-test mode, protect mode, block mode, tabbing, addressable cursor. Microprocessor controlled, programmable underline, line and character insert/delete. "C" version features typewriter-style keyboard. List $950

OUR PRICE $789

920C (with 11 function keys, 6 edit keys and 2 transmission mode keys, List $1030

ONLY $849

HAZELTINE

1500

ONLY

$879

1410 w/numeric keypad, List $900	$749
1420 w/lower case and numeric pad	849
1510, List $1395	1089
1520, List $1650	1389

SOROC

IQ-120

List $995

SPECIAL

$729

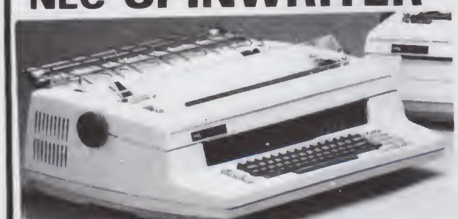
IQ-140 List $1495

SPECIAL $1149

CENTRONICS
PRINTERS

NEW 730, parallel, friction, tractor	**$679**
NEW 737 parallel, friction, tractor	**$849**
779-2 w/tractor (same as TRS-80 Line Printer I), List $1350	1049
702 120 cps, bi-direct., tractor, VFU	1995
703 185 cps, bi-direct., tractor, VFU	2395
704 RS232 serial version of 703, $2350	$1995

NEC SPINWRITER™



Terminal/Keyboard as well as RO Printer Only models available.

CALL FOR PRICES!

PAPER TIGER®

IDS 455 Paper Tiger, List $995	**$895**
w/graphics option, incl. buffer, $1194	$989
TRS-80 cable	45

NEW IDS 460
QUALITY PRINTING AT MATRIX SPEED—LOGIC SEEKING PROPORTIONAL SPACING
w/auto text justification

NEW IDS PAPERTIGER 460 List $1295	$1099
NEW IDS PAPERTIGER 460G List $1394	$1149

TI-810

TI-810 Basic Unit, $1895 . **ONLY**	$1595
TI-810 w/full ASCII (Lower case), vertical forms control, and compressed print	$1795
TI-745 Complete printing terminal with acoustic coupler, List $1695	$1399

ANADEX
DP9500/DP9501 PRINTERS

DP-9500, List $1650	$1349
DP-9501, List $1650	$1349
ANADEX 80-Col. Dot Matrix	$849

OKIDATA
Microline 80 ONLY $599

Tractor Feed Option	$109
Serial interface	$ 99
AXIOM IMP I	$699
COMPRINT 912 w/parallel interf.	$559
912 w/serial interface, List $699	$589
MICROTEK, List $750	$675

Above prices reflect a 2% cash discount (order prepaid prior to shipment). Add 2% to prices for credit card orders, C.O.D.'s, etc. Prices are f.o.b. shipping point. Prices are subject to change and offers subject to withdrawal without notice. **WRITE FOR FREE CATALOG.**

226

MiniMicroMart, Inc.

VISA

master charge

1618 James Street, Syracuse NY 13203 (315) 422-4467 TWX 710-541-0431

Reader Service—see page 210

kilobaud

Peterborough NH 03458

MICROCOMPUTING T.M. LIST OF ADVERTISERS

kilobaud MICROCOMPUTING

*This advertiser prefers to be contacted directly.

For further information from our advertisers, please use the Reader Service card.

READER SERVICE

Please help us to bring you a better magazine—by answering these questions:

I. I am interested enough in a general-interest article to see it in lieu of a technical article.
- ☐ 1. Yes
- ☐ 2. No

II. The following types of general interest articles appeal to me most (check two only).
- ☐ A. Humor
- ☐ B. Interview with users/industry leaders
- ☐ C. Historical
- ☐ D. Essay
- ☐ E. Profile of a business

III. As a reader of Microcomputing, I consider myself a
- ☐ 1. Beginner
- ☐ 2. Intermediate
- ☐ 3. Expert

IV. The articles in Microcomputing are
- ☐ A. Too simple
- ☐ B. Too complex
- ☐ C. Just right

V. I want to see more industry news in Microcomputing
- ☐ 1. Yes
- ☐ 2. No

VI. The following major computer publication gives me the most usable information (check only one)
- ☐ A. Kilobaud Microcomputing
- ☐ B. 80 Microcomputing
- ☐ C. Byte
- ☐ D. Interface Age
- ☐ E. Personal Computing
- ☐ F. Creative Computing
- ☐ G. Dr. Dobb's Journal

VII. I want to see more regular departments in Microcomputing
- 1. Yes
- 2. No

VIII. I bought this magazine on the newsstand because (answer only if question applies)
- ☐ A. I was looking specifically for Microcomputing
- ☐ B. The cover caught my eye

IX. I want to see more of the following kinds of articles in Microcomputing (check two only)
- 1. Business
- 2. Construction
- 3. Education
- 4. Game
- 5. Home application
- 6. Human interest
- 7. Programming technique
- 8. Review
- 9. Scientific

X. Is the advertising content of the magazine a primary reason for your buying Kilobaud?
- A. Yes
- B. No

XI. What type of computer do you own? (Check all that apply)
- 1. Apple
- 2. PET
- 3. TRS-80
- 4. Atari
- 5. Heath
- 6. TI
- 7. Other

XII. If you are not a subscriber, please circle 500.

Kilobaud MICROCOMPUTING • April 1981

Reader Service: Return this card to receive full information on the products advertised in this issue. Refer to the ad. You will find numbers near the logo of each advertiser. Each represents the advertiser's individual Reader Service number. Circle the corresponding numbers on one of the cards on this page, include your name, address & zip, and drop in a mailbox. In 4-6 weeks you'll hear from the advertiser directly. **This card valid until May 31, 1981.**

1	6	11	16	21	126	131	136	141	146	251	256	261	266	271	376	381	386	391	396
2	7	12	17	22	127	132	137	142	147	252	257	262	267	272	377	382	387	392	397
3	8	13	18	23	128	133	138	143	148	253	258	263	268	273	378	383	388	393	398
4	9	14	19	24	129	134	139	144	149	254	259	264	269	274	379	384	389	394	399
5	10	15	20	25	130	135	140	145	150	255	260	265	270	275	380	385	390	395	400
26	31	36	41	46	151	156	161	166	171	276	281	286	291	296	401	406	411	416	421
27	32	37	42	47	152	157	162	167	172	277	282	287	292	297	402	407	412	417	422
28	33	38	43	48	153	158	163	168	173	278	283	288	293	298	403	408	413	418	423
29	34	39	44	49	154	159	164	169	174	279	284	289	294	299	404	409	414	419	424
30	35	40	45	50	155	160	165	170	175	280	285	290	295	300	405	410	415	420	425
51	56	61	66	71	176	181	186	191	196	301	306	311	316	321	426	431	436	441	446
52	57	62	67	72	177	182	187	192	197	302	307	312	317	322	427	432	437	442	447
53	58	63	68	73	178	183	188	193	198	303	308	313	318	323	428	433	438	443	448
54	59	64	69	74	179	184	189	194	199	304	309	314	319	324	429	434	439	444	449
55	60	65	70	75	180	185	190	195	200	305	310	315	320	325	430	435	440	445	450
76	81	86	91	96	201	206	211	216	221	326	331	336	341	346	451	456	461	466	471
77	82	87	92	97	202	207	212	217	222	327	332	337	342	347	452	457	462	467	472
78	83	88	93	98	203	208	213	218	223	328	333	338	343	348	453	458	463	468	473
79	84	89	94	99	204	209	214	219	224	329	334	339	344	349	454	459	464	469	474
80	85	90	95	100	205	210	215	220	225	330	335	340	345	350	455	460	465	470	475
101	106	111	116	121	226	231	236	241	246	351	356	361	366	371	476	481	486	491	496
102	107	112	117	122	227	232	237	242	247	352	357	362	367	372	477	482	487	492	497
103	108	113	118	123	228	233	238	243	248	353	358	363	368	373	478	483	488	493	498
104	109	114	119	124	229	234	239	244	249	354	359	364	369	374	479	484	489	494	499
105	110	115	120	125	230	235	240	245	250	355	360	365	370	375	480	485	490	495	500

Name ____________________

Address ____________________

City ____________________ State ________ Zip ________

Microcomputing • POB 2741 • Clinton IA 52735

BOOKS

Please send me the following Microcomputing products:

Qty.	Catalog #	Title	Unit Price	Total

Add $1 shipping/handling ________

Total ________

Please allow 4–6 weeks for delivery. No C.O.D.s accepted.

Enclosed $________ ☐ Check ☐ M.O.

Bill: ☐ AE ☐ MC ☐ Visa

Card # ____________________ Exp. date ________

Signature ____________________ Interbank # ____________________

Name ____________________

Address ____________________

City ____________________

State ____________________ Zip ________

4/81

Microcomputing • Mail Order Dept. • Peterborough NH 03458

SUBSCRIPTION

MICROCOMPUTING subscribers save $10 off the newsstand price.

- ☐ New subscription ☐ Renewal
- ☐ 1 year —$25
- ☐ 2 years—$38
- ☐ 3 years—$53

Enclosed $________ ☐ Check ☐ M.O.

Bill: ☐ MC ☐ Visa ☐ AE ☐ me

Card # ____________________ Exp. date ________

Signature ____________________ Interbank # ____________________

Name ____________________

Address ____________________

City ____________________ State ____________________ Zip ________

Canada—$27, 1 year only, US funds

Other foreign—$35, 1 year only, US funds

314B9

Microcomputing • POB 997 • Farmingdale NY 11737

kilobaud MICROCOMPUTING™

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 217 CLINTON IA 52735

POSTAGE WILL BE PAID BY ADDRESSEE

kilobaud MICROCOMPUTING T.M.

POB 2741
Clinton IA 52735

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO 1024 PETERBOROUGH NH 03458

POSTAGE WILL BE PAID BY ADDRESSEE

kilobaud MICROCOMPUTING T.M.

Subscription Dept.
POB 997
Farmingdale NY 11737

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 1024 PETERBOROUGH NH 03458

POSTAGE WILL BE PAID BY ADDRESSEE

kilobaud MICROCOMPUTING T.M.

Peterborough NH 03458

Att. Mail Order

LETTERS

(from page 29)

with dummy data to obtain the proper entry. In either case, this would mean lost memory, speed and efficiency.

This principle of indexing into tables is an important and extensively used application in assembly-language programming. In a higher-level language, this would be similar to an array. The efficiency of indexing using zero numbering is the same whether there are 77, 128 or even two elements in a table.

Secondly, examine any assembly language and see how many instructions refer to zero. Each of these instructions calls for the processor to do some function (jump, return, call, etc.) if equal to 0, not equal to 0, greater than 0, less than 0. Here, zero numbering comes into play again. If our program was to read all 77 tracks on a disk, it is very easy to exit the read loop after it has read track 0 by using one of these conditional instructions. There are ways around this, but usually at a loss of speed, efficiency and usually not conforming to structured programming.

Thirdly, Mr. Piescik's point is a valid one. With zero numbering, memory is conserved, since more items can be described with the same number of bits. Even though the memory saving may be small by itself, throughout an entire system, it can become a significant factor.

To illustrate how all three points can fit together, suppose our system can transfer a character from any one of 32 CRTs to any one of 32 printers (assume a four-bit processor). Already the illustrious salespeople can sell one additional CRT and one additional printer with the same software (point #3). Our program starts with the last CRT (#31) and can quickly index into a table for the appropriate routing and processing (point #1). After CRT #0 is done, a simple zero conditional instruction takes us out of the program (point #2).

While it is confusing to refer to the first item as item #0 and may not be necessary when using higher-level languages, it is definitely *not* something that someone invented just to confuse novices. Zero numbering is as much a part of programming as the binary system.

William McGarry
Stratford, CT

In the Switch of Time

The construction article in the July issue ("A Hexadecimal Front Panel for Z-80 Systems," John D. Ciana, p. 148) could not have come at a better time. I had been planning a homebrew Z-80 system for some time, and I was working on a front panel design. The cost of my design would run about $120, would have less capability than John Ciana's configuration and would require a giant leap backward to the prehistoric Altair-style toggle switch approach. By contrast, Ciana's circuit would cost less and provide hardware single-step and breakpoint.

The circuit seemed satisfactory for the most part, but instead of buying ten of those expensive hex decoder chips, I decided to multiplex the ten display digits. As a result I saved money, not only on decoders, but also on power consumption (I need one less 5-volt regulator). The multiplex circuit is shown in Fig. 1.

Bruce Malmat
Silver Spring, MD

False Advertising

I am usually very unhappy concerning false advertising, such as appears in your ad on p. 230 of the December 1980 issue. I refer to the Library Shelf Boxes which are advertised to ". . . hold a full year of *Kilobaud Microcomputing*."

I have news for you. The year should have ended after the October issue.

Keep up the good work. I'd love to fill one up in six months!

John W. Warne
Gainesville, FL

Wari Is a Winner

I have just loaded the Wari game by L. D. and P. H. Stander, published in *Kilobaud Microcomputing*, November 1979, p. 118. It was very easy to adapt to Microsoft BASIC and run it on my Sorcerer computer.

The reason I am writing is to thank you for publishing it. It is one of the more enjoyable games I have seen because it is somewhat challenging if played against a computer or a human opponent, and it runs fast and loads in less than 4K of BASIC.

Jacob L. Zar
No. Andover, MA

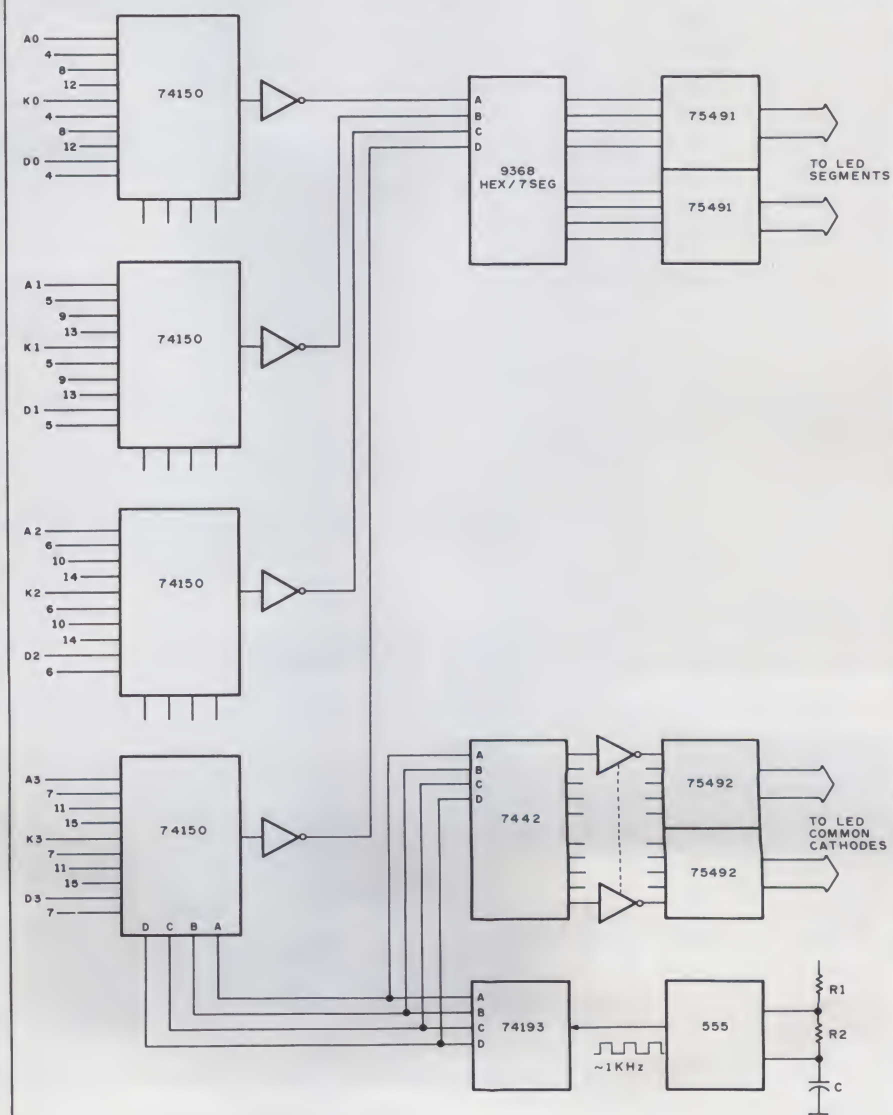


Fig. 1.

NEW PRODUCTS

Edited by Dennis Brisson

Model III Mini-Disk Storage System
PMC-80 Expander
IEEE 488 Acoustic Modem
Apple Laboratory Interface Card

Four Mini-Disk Drives for TRS-80 Model III

TRS-80 Model III owners can now add up to four drives to their computers, and thus increase on-line formatted storage capacity to almost 1.5 megabytes, with mini-disk (5-1/4 inch) storage systems from Percom Data Company, 211 N. Kirby, Garland, TX 75042. Systems come with either 40- or 80-track drives that are rated for both double- and single-density operation. Both drive models are capable of "flippy" operation, so that diskette storage can be twice the on-line capacity. The first two drives mount inside the Model III; the third and fourth drives mount external to the computer. The initial internal drive system sells for $749.95 (40-track) and $914.95 (80-track). Reader Service number 474.

The PMC-80 and peripherals.

Disk-Based System For PMC-80

The model EXP-100 expander for the PMC-80 provides 32K memory expansion, minifloppy disk interface for four drives, Centronics parallel printer interface, RS-232C interface and an S-100 bus to provide additional interface capabilities. With the Expander, the PMC-80 is capable of a number of configurations ranging from a simple, low-cost Level II tape-based system to a fully implemented word processing or business system utilizing the same software as was developed for the TRS-80 Model I computers. It will run TRSDOS software as well as other operating systems designed for TRS-80 compatibility such as NEWDOS and VTOS. Prices are $410 without memory and $245 for 16K or $295 for 32K memory board.

The model PIF-40 plug-in printer interface permits any printer with a Centronics-compatible interface to be connected to the PMC-80 without the need for the Expander. Price is $95.

Personal Micro Computers, Inc., 475 Ellis St., Mountain View, CA 94043. Reader Service number 472.

Percom's mini-disk drives for the Model III.

Acoustic Modem

The Star 488 is the newest

LOWEST PRICE - BEST QUALITY

NORTH STAR

North Star Horizon 2

2-5¼ Disk Drives
32K Double Den
Factory assem. & tested
Factory guaranteed
List 3095

only **$2274**

POWERFUL NORTH STAR BASIC FREE
SUPERB FOR BUSINESS & SCIENCE

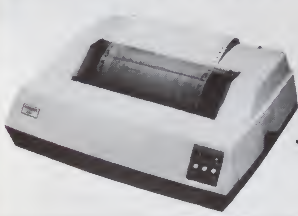
FACTORY ASSEMBLED & TESTED	LIST	ONLY
HORIZON-1-32K-DOUBLE DEN	$2695	$1980
HORIZON-2-32K-QUAD DENSITY	3595	2674
HORIZON-2-64K-QUAD + HARD DISK	9329	7149
HORIZON RAM ASSM	16K = $389.	32K = $579
HORIZON RAM KIT	SALE! 16K = $314	32K = $469
HORIZON DISK DRIVE SALE DOUB DEN	SAVE!	315
NORTH STAR HARD DISK 18 Mb	4999	$3929
PASCAL-PLUS 14,18 OR 36 DIGIT PRECISION		249

SUPERBRAIN
ZENITH

SUPERBRAIN QD 64K
List $3995 only $2995

Z-89 48K
List $2895 only $2299

TERMINALS Z-19 $725
INTERTUBE III only $725
DIP-81 PRINTER only $395



MICROTEK $675

NEC PRINTER $2569
TRACTOR, THIMBLE, RIBBON

InterSystems

ITHACA INTERSYSTEMS 2A

Z-80A CPU 4 MHz
64K Dynamic RAM
Front panel
V I/O—with interrupts
FDCII Disk Controller
20 slot motherboard

List $3595 You Pay Only **$2795**

PASCAL/Z + THE FASTEST PASCAL $375
GET READY FOR ITHACA'S Z-8000
8086 16 BIT CPU & SUPPORT CARD SEATTLE $575

MORROW 8" DISK

DISCUS 2D + CP/M® 600K ONLY $938
DISCUS 2 + 2 + CP/M® 1.2 MEGA B. $1259
ADD DRIVES 2D = $650 2 + 2 = $975
2D-DUAL + CP/M® + MICROSOFT BASIC = $1555
$2293 VALUE OVER 32% OFF!
WHILE THEY LAST

MORROW HARD DISK
26,000,000 BYTES!!
LIST $4995 ONLY $3995
CP/M® IS INCLUDED!

SAVE ON MEMORY AND PROGRAMS

SYSTEMS MEMORY 64K A & T 4mHz $599
SYSTEMS MEMORY 64K BANK SELECT $789
CENTRAL DATA 64K RAM $599
ITHACA MEMORY 8/16-bit 64K $845
SEATTLE MEMORY 8/16 BIT 16K 4Mhz $275
SSM KITS Z-80 CPU $221
VIDEO BRD V83 4Mhz $412
ANADEX PRINTER DP-9500-1 $1349
CAT NOVATION MODEM $169
TARBELL DISK CONTROLLER DD $445
ECONORAM 2A 8K ASSM $179
NSSE 1-22 & P01 TERRIFIC PROGRAMS ONLY $10. EACH
NORTHWORD 294 MAILMAN 234
INFOMAN $364
TARBELL COMPUTER-PHONE

RCA-COSMAC VP-111 99 RCA-COSMAC VP-711 $199
COLOR! RAINBOW 385
SPECTRUM $289
EZ-80 Machine Language Tutor $25
EZ-CODER Translates English to BASIC $71
ECOSOFT FULL ACCOUNTING PKG $315
BOX OF DISKETTES $29
SECRETARY WORD PROCESSOR The Best! $99
TEXTWRITER III Book Writing Program $112
GOFAST NORTH STAR BASIC Speeder Upper $71
Which Computers are BEST? BROCHURE FREE
North Star Documentation refundable w/HRZ $20

AMERICAN SQUARE COMPUTERS BEATS ADV. PRICES

American square Computers

56

919-889-4577 KIVETT DR. JAMESTOWN N.C. 27282 919-883-1105

® CP/M is a registered trademark of Digital Research, Inc.

Livermore's Star modem.

in a series of 300 bps acoustic modems from Livermore Data Systems, Inc., 2050 Research Drive, Livermore, CA 94550. It includes the direct IEEE 488 interface necessary for use with Hewlett-Packard, Tektronix and Commodore PET computers.

The Star 488 features crystal-controlled frequency generation, a convenient self-test mode and exclusive triple-seal cups that lock the handset into acoustic isolation and adapt to most handsets used throughout the world. This $10.75 \times 4.5 \times 2$-inch unit weighs two pounds and is compatible with all 103-type modems. Reader Service number 471.

Interactive Microware's APPLAB for the Apple II+.

Laboratory Computer System

The APPLAB interface card, designed for the Apple II+ computer, can be used to control or collect data from most scientific instruments, such as spectrophotometers, chromatography systems, pH meters, strip chart recorders and temperature controllers. It features 12-bit D/A and A/D converters with jumper-selectable ranges of ± 0.5 V, ± 1.0 V, ± 2.0 V and ± 4.0 V. The digital subsystem features eight bits each of input and output, versatile handshaking signals, interrupt circuitry and TTL-compatible signal levels. A 32-bit real-time clock displays time in hours, minutes and seconds and permits timing of events to an accuracy of 0.1 second. Two 16-bit timers may be configured as an interval timer, pulse counter, pulse generator, square wave frequency generator or shift register. Price is $495.

Interactive Microware, Inc., PO Box 711, State College, PA 16801. Reader Service number 482.

Surge Suppressors

Kalglo's Spike-Spiker.

Kalglo Electronics Co., Inc., Colony Drive Industrial Park, 6584 Ruch Rd., East Allen Township, Bethlehem, PA 18017, has added two new models to its Spike-Spiker series:

Spike-Spiker Mini I features two grounded outlets and plugs directly into the 120-volt wall outlet. It protects against most power-line transients.

The Spike-Spiker Mini II, in addition to protecting against power-line transients, also provides rf "hash" filtering of conducted interference between the computer or sensitive instruments from motorized or "noisy" equipment in the vicinity of the computer. These $3 \times 4 \times 1$-1/2 inch units weigh less than 1/2 lb. and are prewired and ready to use. Reader Service number 476.

The Super Isolator from Electronic Specialists.

Computer Pollution Control

Stamp out electrical pollution, which can often cause erratic computer operation or even expensive hardware damage, with the Super Isolator from Electronic Specialists, Inc., 171 South Main St., Natick, MA 01760. Incorporating heavy-duty spike/surge suppression, the Super Isolator features three individually dual-Pi filtered ac sockets to eliminate equipment interactions and control disruptive, damaging power line pollution. It will control pollution for an 1875 watt load. Each socket can handle a 1000 watt load. Price is $94.95. Reader Service number 470.

Insertion and Extraction Tools

OK Machine and Tool Corporation, 3455 Conner St.,

AID's microprofile keyboard design.

Bronx, NY 10475, has introduced its new MDD series of DIP IC dispensers for MOS and CMOS, as well as standard devices. Each channel accepts any standard IC shipping tube, and can accommodate any standard IC from 2–42 pins on .300, .400 or .600 centers. Adjustable guides position each IC individually for easy extraction, and simple gravity feed assures reliable deposit of the next IC into extraction position after the previous IC is removed. Available in 1-, 5- and 10-channel versions for $21.85, $83.43 and $160.45, respectively. Reader Service number 473.

25 Watt Plug-In Power Supply

Now a wall plug-in power supply that delivers up to 25 watts of regulated dc power is available from Ault, Inc., 1600H Freeway Blvd., Minneapolis, MN 55430. Two versions are offered for microprocessor-based systems: single output (+5 V at 5 amperes) and triple output (+5 V at 2.5 amperes and ±12 V at .2 amperes). There is also a single-output unit (48 V at .5 amperes) for telephone applications. These new power supplies can be plugged into any conventional wall outlet. Reader Service number 477.

New Keyboard Design

The MK 058 is a 58-key sealed keyboard with a microprofile button construction (0.4 inch), which allows improved packaging capabilities for portable and desk-top applications. It provides a crisp, tactile feedback to the operator, and features custom legend sheets and keytop colors. Legends are placed on the bezel to provide a more advanced styling or can be added to the buttons to allow more than one operating mode for the keyboard.

Advanced Input Devices, PO Box 1818, Coeur d'Alene, ID 83814. Reader Service number 483.

Letter-Quality Printer

The Sellum I is an intelligent, letter-quality printer that features automatic bidirectional printing, switch/software-selectable baud rate and logic-seeking capabilites that allow a maximum of 700 words per minute. The 650-character buffer and optional 16K buffer for printer spooling allow simultaneous data entry and printing. An optional front panel and Diablo-compatible software are available.

The Sellum I is compatible with any computer utilizing RS-232-C serial, Centronics-type or 20 uA current-loop interfaces, and will support both keyboard and additional firmware. Price is $3495.

Intersell, 465 Fairchild

Ault's plug-in power supply.

Reader Service—see page 210

Mimic's plug-in speech processor for the TRS-80.

Drive, #214, Mountain View, CA 94043. Reader Service number 480.

Digital Speech Processor

The Mimic Speech Processor converts speech signals to a digital bit stream for computer storage or automatic speech recognition purposes. It also reconstructs the digital speech representations to analog form for reproduction through an available speaker. Since both the speech encoder (input) and decoder (output) functions are located in a single unit, the system can also be used for speech communications applications without a computer. Speech data rate is user selectable. Several versions are available: a bare board for user assembly (under $20), a fully assembled and tested module ($79), a general-purpose plug-in system which connects directly to a parallel I/O port on most computers ($149) and a complete TRS-80-compatible plug-in version ($169).

Mimic Electronics Co., PO Box 921, Acton, MA 01720. Reader Service number 481.

The DSAIM-65 from Applied Business Computer.

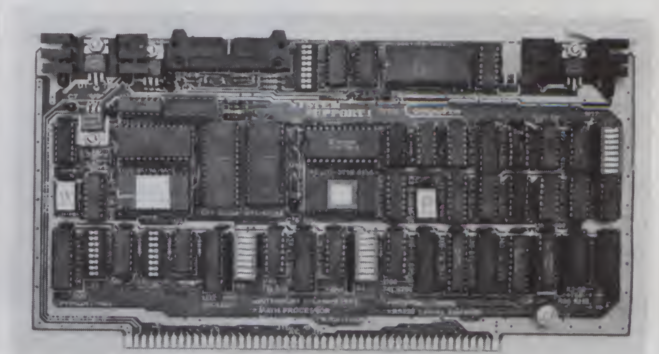
AIM-65 Development System

Applied Business Computer Co., 707 S. State College Blvd., Suite G, Fullerton, CA 92631, has recently announced the AIM-65 Development System. Designated the DSAIM-65, this development system for the 6500 microprocessor family features an 80-character × 25-line display, 32K expandable to 64K RAM, dual single-sided double-density minifloppy drives with over 280K of mass storage on-line expandable to over 1.1 megabytes, six-card-slot expansion board and a Centronics-type printer interface. Software includes the ADOS disk operating system, BASIC high-level language, assembler, editor and debug monitor. Reader Service number 475.

S-100 Multifunction Board

System Support 1, a multifunction board for S-100 microcomputers, provides sockets for 4K of EPROM or RAM, as well as a socket for an optional math processor, two interrupt controllers, real-time calendar/clock, three 16-bit interval timers and a full RS-232 serial channel with software-selectable baud rate. If micropower RAM is used in one of the memory sockets, it can also run from the same battery used to back up the clock in the event of power failure. Price is $395, assembled and tested.

CompuPro, Building 725, Oakland Airport, CA 94614. Reader Service number 478.



CompuPro's System Support 1.

Software for the TRS-80 Color Computer
PET Educational Programs
CP/M-Based Editor
Heath Inventory Management System

CP/M-Based Editor

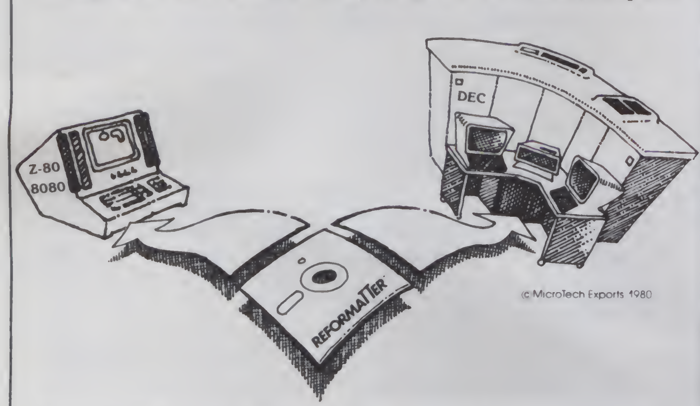
CP/M-based text processing is easy to learn with Mince, a full-screen editor that uses a mnemonic control-character command set, allowing all textual characters to be entered directly into the document without confusing input and edit modes. With over 80 commands, Mince can edit more than one document to another and display two documents on the screen at once. It features automatic paragraph fill and commands which deal with words, sentences, paragraphs and user-defined text regions, as well as lines and characters. Price is $125.

Mark of the Unicorn, PO Box 423, Arlington, MA 02174. Reader Service number 468.

CP/M-DEC Reformatter

Now your Z-80- or 8080-based microcomputer can exchange data files with DEC computers using the floppy disk as the transfer medium. The Reformatter diskette conversion program runs under the CP/M operating system and reads and writes floppy disks in the DEC RT-11 format. You can transfer data files bidirectionally and alter any of the fields in the DEC RT-11 directory. Reformatter also lists the DEC directory and displays the unused area of the disk. A squeeze function, which allows a "fragmented" DEC diskette to be packed into a continuous data area, is available. Price is $195.

MicroTech Exports, 467 Hamilton Ave., Palo Alto, CA 94301. Reader Service number 491.



MicroTech Exports' Reformatter allows data exchange between Z-80 or 8080-based micros and DEC equipment.

CP/M for North Star

North Star CP/M 2.2 is a fully compatible version of CP/M for both floppy and hard disk systems for Horizon system users. This monitor control program for microcomputer system development uses both North Star 5-1/4-inch floppy and 14-inch hard disks for storage. It provides a general environment for program construction, storage and editing, along with assembly and program checkout facilities. This new release consists of a single diskette which works with double density, quad capacity and hard disk systems, and features a unique sequential file access capability.

North Star Computers, Inc., 1440 Fourth St., Berkeley, CA 94710. Reader Service number 488.

Newtalk

The machine-language programmer now has a convenient way of checking memory contents against a printed program listing with the Newtalk program for 6800 and 6809 systems. It is a completely relocatable utility which does a byte-by-byte memory dump of a selected memory area, and prints the output on the screen as well as reading it out through a loudspeaker. A speaker/amp, signal tracer or telephone amplifier is required to amplify and convert the computer output to sound. Price for disk or cassette is $35.

Star-Kits, PO Box 209, Mt. Kisco, NY 10549. Reader Service number 484.

Heath/Zenith Business Software

The Inventory Management system lists over 1100 items, which you can add, delete, change and display, and 17 fields of information for each item for use in small-business operations. Direct on-line sales/returns and orders/receipts entry provide automatic on-hand and sales history adjustment. Status, sales history, reorder, physical reconciliation and user-selectable summarization reports are available. It requires the Heath/Zenith H8/H19 or H-89 with HDOS operating system, Microsoft BASIC, 48K RAM, two 5-1/4-inch disk drives and a 132-column line printer. Price is $199.95.

XtraSoft, Inc., PO Box 91063, Louisville, KY 40291. Reader Service number 467.

Apple Apartment Management

The Landlord provides property owners and managers with listings of apartments, residents and past residents, as well as reports on vacancies, lease expirations, intents to vacate and resident payments. It can handle apartment properties of up to 400 units. Records of disbursements and other financial transactions are maintained by the system, and a monthly property analysis statement is produced. Resident charges and payments, as well as security and pet deposits, returned checks and overpayments are handled by the system. A balance report allows expedient followup of delinquent residents. It requires an Apple II with 48K RAM, two disk drives and either a Silentype or Centronics 779 printer. Price is $795.

MIN Microcomputer Software, Inc., 5835-A Peachtree Corners East, Norcross, GA 30092. Reader Service number 490.

Chinese Lessons From Apple

Do you have a yen to learn the Chinese language? In 11 easy-to-use lessons you can master Chinese greetings, times, seasons, numbers, foods and other commonly used terms (200 in all). Color, graphics and sound are used in each lesson, and memory aids, meanings and pronunciations are presented with the Chinese characters. The program features drill and practice under user control and exercises with exam scoring. Chinese Lessons is available for the Apple II 48K in Applesoft with a single disk drive on a reversible 5-1/4-inch diskette. Price is $29.95, plus $1.50 shipping and handling.

Computer Translation, Inc., Department EMG, Box 7004 University Station, Provo, UT 84602. Reader Service number 463.

Atari Exercise Program

Health-conscious computer owners can now stay in shape with the Atari Personal Fitness program. A series of questions and answers provides information the computer uses to formulate a customized exercise program. The computer then displays an exercise figure and counts out the routine. At the end of each session, progress is recorded and charted, then revised to meet physical fitness goals. The program is on cassette tape, requiring a minimum of 16K RAM and an Atari 410 Program Recorder.

Atari Consumer Division, 1265 Borregas Ave., PO Box 427, Sunnyvale, CA 94086. Reader Service number 489.

6502 Assembler Package

HESBAL is a full-featured 6502 assembler for your PET/CBM. It leaves over 1200 bytes free (8K PET) for your use and requires one tape or disk drive. Features include label size of six characters or more, several pseudo-opcodes and over 25 error messages.

HESEDIT is a full-screen editor which can be used to prepare source for HESBAL, or can be used by itself for maintaining data files. Features include repeating keys; editing 22 lines at a time; inserting, deleting and duplicating lines; and scrolling forwards and backwards by any amount. Loads on old or new ROMs. Prices are $23.95 for cassette and $26.95 for diskette.

Human Engineered Software, 3748 Inglewood Blvd., Room 11, Los Angeles, CA 90066. Reader Service number 492.

Color Computer Software

The following new utility software has been introduced for the TRS-80 Color Computer:

—Renumber provides the ability to load a program, renumber it and then save it.

—Squeeze will compress BASIC code to utilize minimum memory.

—Merge allows two separate programs on cassette to be concatenated in memory and then saved. Cassette price is $19.95.

Mint Software, 6422 Peggy St., Baton Rouge, LA 70808. Reader Service number 494.

—The CBUG Monitor tape features 19 commands and allows you to
—examine or change memory using a formatted hex display
—save areas of memory to cassette in binary
—download/upload data or programs to a host system
—move the video display page throughout RAM
—send or receive RS-232 at up to 9600 baud
—use your Color Computer as an intelligent peripheral for another computer, a color display or a 6809 program development tool. Price is $29.95. A disassembler that runs on the Color Computer and allows you to generate your own source listing of the BASIC interpreter ROM is also available for $49.95.

349

The Micro Works, PO Box 1110 Del Mar, CA 92014. Reader Service number 495.

PET Educational Software

The following educational software for the PET computer has recently been released by Teaching Tools, Microcomputer Services, PO Box 12679, Research Triangle Park, NC 27709:

—Addition and Subtraction programs suitable for grades one through six and for special education classes. Immediate feedback is given, with graphics reinforcing correct responses. You can set the number and level of difficulty of the problems, the time limit and the number of attempts allowed on each problem.

—Letters and Numbers program for young children first learning to use the computer. Using large letters and numbers, it tests for matching of one or more items, completing sequences and filling in missing items.

Standard & Poor's system for the TRS-80.

—Match Game program useful for memory building and for putting lesson reviews into a game format. One to four players can participate in matching shapes and synonyms and answering math problems, or single players can play against the computer.

Each program, which costs $20, requires at least 8K and 1.0, 2.0 or 4.0 ROM. Reader Service number 485.

Stock Analysis Program

Now you can duplicate the professional investment strategies used in the financial community with Stockpak, a complete stock analysis and portfolio management package. Standard & Poor's system provides for evaluating and managing a stock portfolio of up to 100 securities with as many as 30 transactions on each issue. You can also analyze 900 New York and American exchange and over-the-counter common stocks and generate reports to guide investment decisions. In addition, you can record buy and sell transactions, price and dividend information and stock splits and instantly retrieve this information for record keeping and tax purposes. It is designed for use with the Model I or III TRS-80 32K business computer system. Price is $49.95.

Radio Shack, 1800 One Tandy Center, Fort Worth, TX 76102. Reader Service number 498.

Mailing List Data Base System

Now you can organize your mailing lists, personal address book functions or mail lists based on dates with Maillist for the TRS-80 Models I, II and III. It uses an ISAM-based file system to minimize disk access times. A data base consists of a master file and up to four index files, which allows the primary file to be maintained in sorted order by name, zip code and up to two dates. Maillist allows you to specify up to 30 attributes, which are used in record selection when generating reports or mailing labels. It supports nine-digit zip codes and three-digit state codes. Prices are $75 for the Models I and III and $150 for the Model II.

Racet Computes, 1330 N. Glassell, Suite M, Orange, CA 92667. Reader Service number 486.

Audiophile Library System

Audiophile Library System (ALS) is for the serious music lover who wants to inventory his large collection of records, tapes or cassettes on disk. Each selection added to the file is stored by record number, tape number, *title*, composer, artist(s), conductor, orchestra and source. A specific selection or all selections with a common attribute can be instantly retrieved and displayed on the screen; it is not necessary to use the complete nomenclature of the item to be found. Designed for the TRS-80 Models I and III with TRSDOS, ALS is for single-disk or two-disk format. Price is $19.95, plus $1 shipping and handling.

GB Associates, PO Box 3322, Granada Hills, CA 91344. Reader Service number 496.

TRS-80 Speedup Program

Faster is a software speedup utility program for most TRS-80 Model I, Level II and disk BASIC programs. It analyzes executing BASIC programs, then displays or prints information that enables you to make a simple change (usually just one line). Resulting execution times are typically reduced 10–50 percent. No hardware modifications are involved.

Prosoft, Box 839, No. Hollywood, CA 91603. Reader Service number 493.

✓Reader Service—see page 210

OSI Book Strikes Familiar Chord
Getting Along with a Computer in Business
Raising Computer Consciousness
Intro to LoRes Graphics

VIP, Very Important Programs for Ohio Scientific

S. Roberts
ELCOMP Publishing, Inc.
3873L Schaefer Avenue
Chino, CA 91710
Softcover, $5.95

Oh boy! Very important programs for Ohio Scientific computers! There should be lots of utilities: line renumberer, branch finder, variable cross-reference table maker, program compactor, perhaps a simple word processor, a fix for the error code display, maybe an enhanced line editor.

Well, one person's "very important" is another's "some useful." Not one of the above programs is in the book. However, the loss to OSI users may be a gain for others. All of the 45 programs are in BASIC and most are not at all limited to OSI machines.

Ten of the programs are games, mostly of medium length (about 100 lines). Although I have not seen these particular games before, they follow familiar models.

A section called "Useful Programs" contains such things as a memory test and a hex dump. The section on personal utilities has short programs like "Dollar Converter" (to German marks) and "Speed vs. Gasoline Consumption."

The longest section, and the one containing the highest-quality programs, is called "Useful Math Routines." Among them is a set of matrix programs of various types (for matrices up to 5 x 5 in size), Simpson's rule for integration, histograms and regression analysis. The longest of these is a Prey-Carnivore coupled differential equations program, about 150 lines long, using the Runge-Kutta method.

Now, I must comment on a discovery, but I don't quite know how to put it. As I was reading page 22, my eye caught the sentence "Thoughts of a BASIC statement X=USR(Y)(Z) should be jumping into your head about now." This distinctive idea is word-for-word out of *The (Real) First Book of OSI*, by Jim Williams and George Dorner, or its precursor, both published by Aardvark Technical Service. The list of addresses this sentence is embedded in seem also to be from the precursor, but my copy of *VIP Programs* is defective here and I can't be sure.

In any case, this discovery took the edge off the surprise I experienced when I reached page 81, entitled "POKE AND PEEK," and found it was word-for-word from my own (copyrighted) book *All About OSI BASIC in ROM*, first edition, pages 16 and 17. If imitation is the most sincere form of flattery, being copied without proper attribution generates a somewhat different emotion. I am glad that information about the OSI complex is being widely distributed, but would hope that all concerned give proper credit to the originators for material used.

"But," you say, "your work is protected by copyright." Yes. For that matter so is *VIP Programs*, by Mr. Winfried Hofacker. All a copyright does is provide a license to sue. I am not at war with anyone; I just want to produce good and useful output and expect that reasonable rewards will be returned by honest users of my work.

In the same vein, the writer(s) of these programs now have a vehicle, in this book, that disseminates their work to potential users. Because of the scattered subject matter of these programs, I cannot recommend it at this price ($9.95) to any one user. But it should be stocked at computer stores, and browsers may find some programs that meet their needs and make the book worth the price.

Edward H. Carlson
Okemos, MI

The Small Computer in Small Business: A Guide to Selection and Use

Brian R. Smith
The Stephen Greene Press, 1981
143 pp., $12.50

There are over ten million small business owners in the United States, and many of them would like to know whether a small computer could save them money and, if so, how to go about getting one. Any book that can successfully answer these questions could sell not only a lot of copies, but a lot of computers as well.

Brian Smith's *The Small Computer in Small Business* almost makes it. Subtitled "A Guide to Selection and Use," the book is written not for the experienced business person but the computer novice. Its nine chapters include not only an introduction (albeit brief) to computers, peripherals and software, but key points for judging cost effectiveness, planning for the computer's arrival and—should all else fail—doing without.

There is also a list of probing questions to ask other computer owners (and eventually the salespeople), along with warning symptoms of that dread disease, computer fever. A six-page glossary, conveniently keyed to boldface in the text, defines many of the common terms of computerese.

Author Smith is a business consultant and, not unexpectedly, the best chapters in the book relate to business applications. The 38-page discussion of packaged programs, ranging from general ledger through payroll to word processing, is liberally illustrated with sample input/output as well as with pages reproduced from typical software manuals. It is hard to imagine a *better* introductory treatment.

Because the author believes that any

✓Reader Service—see page 210

discussion of specific hardware is already obsolete on day of publication, he confines his discussion to general characteristics and criteria for choice. Alas, the naive reader is likely to emerge without much idea of the difference between IBM's System/3 and the products down at the local computer store (indeed, nearly all the equipment pictured in the book is traditional mini, not microcomputer, gear).

The selection of topics also is a bit odd in some places. It is doubtful that the average business person desires or needs to know the details of a half-adder circuit. Why is the language C mentioned when RPG, more common in the business environment, is not? Winchester technology, program security, licensing agreements, the limitations of BASIC as a business language and the usefulness of source code for program modification are not mentioned.

Finally, the author's veneer of computer expertise wears thin at times. Definitions of byte, bus, add time and macro; the number of sectors on a floppy disk and the number of bits in 64K; and an example of FORTRAN code are all either wrong or misleading.

Within these limitations, however, this book has a lot to recommend it. Combined with a subscription to *Microcomputing* and visits to a few good computer stores, it could help a small business owner make the right choices.

Jon Kapecki
Rochester, NY

Computer Consciousness: Surviving the Automated 80s

H. Dominic Covvey and
Neil Harding McAlister
Addison-Wesley Publishing Co., 1980
Softcover, 212 pp., $5.95

This unpretentious little volume, part of Addison-Wesley's Joy of Computing series, is a pleasant change of pace from the oh-wow intros currently stuffing the book stalls.

The authors avoid the high-tech hard-sell approach so commonly adopted by today's hyper-effervescent computer futurists in favor of a more reasoned and cautious perspective. Rather than discuss the boundless and fantastic potentials of the computer, they focus on its limitations, and thus give a much more realistic idea of what the computer can and can't do.

The book tackles the problem of over-enthusiasm early. "If we think of the computer as the ultimate vehicle to transport us into the future, then we may blindly climb aboard this technology as an agent of progress without knowing precisely where we want it to take us," they warn. "If we feel that the computer represents the pot of gold at the end of the scientific rainbow, then its existence will soon become an end in itself, and the focus of work that requires assistance from automation will shift insidiously from the original topics of investigation to research on the computer system itself. If we covet the computer as a modern philosophers' stone capable of transmuting base data into valuable information without human thought, then we will waste our time on fruitless electronic alchemy. If the computer is in our minds a conspicuous status symbol of the scientific and business communities, then we will spend a lot of money on meaningless trinkets. If we view the computer as the ultimate toy, then in childish glee we will play with this technology, frittering away time and energy on 'gee-whiz' applications of no real significance."

Later in the book, the authors draw upon the history of technology to point out that we have a tendency to be overly optimistic. They recall the days after World War II when the popular press provided us with visions of chrome-and-plastic cities in the clouds, and counsel that we don't make the same mistake with computers.

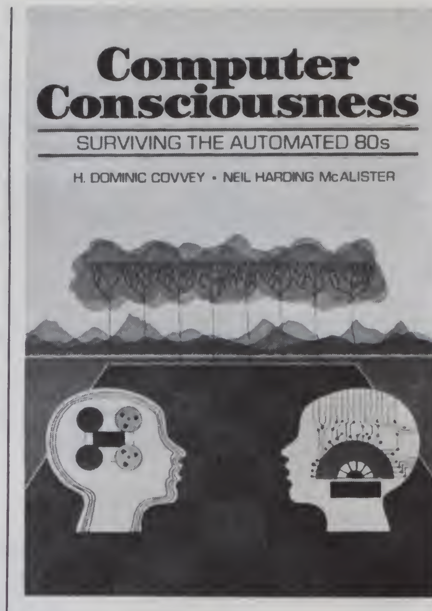
"In system development the tendency of the human mind to leap in a single bound from lowly reality to an idealized concept without considering the boundary between them is a reef on which many a computing project has foundered," they say.

The authors use their reflections as a backdrop for a clear and concise discussion of what a computer is and how it works. They aim their material at the nontechnical computer user, covering all of the usual ground—hardware, software, communications devices, programming languages and so on. The book contains no complicated schematics, but includes a number of simple and informative charts and illustrations.

One of the most helpful chapters is entitled "The System in Situ," subtitled "Computer Systems in the Human Environment." Here the authors take the reader step-by-step through the process of setting up a computer system, again emphasizing a thorough understanding of a computer's real abilities.

"It is not easy to implement any worthwhile computer system," they caution. "A rapid and painless transition may mean only that the computer system being introduced is so unimportant that it has no discernible effect on its environment."

Seasoned computerists won't have much use for this book, but it is worth the time of anyone who is thinking about getting a computer, or who would like to take the first steps toward computer literacy. It would make an excellent high school or college text for any number of nontechnical sociology, science and communications classes.



The computer world could use a few more books written by people who aren't awed by the wonder of it all. *Computer Consciousness* is an excellent model.

Eric Maloney
***Microcomputing* staff**

Introduction to Low Resolution Graphics

Nat Wadsworth
Scelbi Computer Consulting, Inc.
Elmwood, CT, 1979
Paperback, 77 pp., $9.95

Many microcomputers have low-resolution graphics. Yet, their owners don't know how to use this ability, or even that it exists. For these people, this book is a clear, concise and lively introduction.

Wadsworth starts with the fundamentals. Using a hypothetical 16 × 16 display matrix, he shows how the display is divided into tiny squares (pixels), which can be illuminated and turned off independently by a program. He then explains how Cartesian coordinates can be manipulated and used with graphics displays. Finally, he devotes chapters to creating simple shapes, drawing a line between any two points, and making the computer display what the computerist wants it to.

The book explains, in detail, how to control the graphics on the Apple, TRS-80 and PET using BASIC. Wadsworth initially supplies program listings for all three computers, later including only Apple listings along with a brief explanation of how they can be modified for the other two systems.

Chapter five, "A Graphics Library," includes a program that draws a winking, smiling, pointing animated clown. Wadsworth shows how to add sound, so that the clown beeps and buzzes as it winks and opens it mouth. In another example, a series of subroutines draws the individ-

ual cards in a deck. These subroutines, in combination with some of the BASIC card games now available, would make excellent game programs.

The book includes, as its final sample program, an animated football game, similar to Mattel's handheld one. The players are shown as stick figures instead of as blinking lights. They run, have audible footsteps, open their mouths and growl, and are squashed when tackled. Wadsworth provides the complete program listing for the Apple, and several tables to help TRS-80 and PET owners modify it. If you don't want to work out the modifications, you can send in a coupon for the TRS-80 and PET listings. You can also get an Apple program to draw the colorful windmill scene on the book's cover.

My primary criticism of the book concerns the lack of modified ready-to-run program listings for the PET and TRS-80. Since the book is an introduction, why is the reader expected to be able to modify a program he might not understand?

TRS-80 owners need make only minor modifications. But the commands for controlling graphics on the PET are very different. In addition, if you use the method explained here (using each character cell as one pixel), the resolution of the PET is only 40 × 25, compared to 128 × 48 for the TRS-80 and 40 × 40 for the Apple. Wadsworth could have used PET's block graphics set, in which each character cell is divisible into four individually controllable blocks, to increase its resolution to 80 × 50.

Perhaps a program to do this would have been beyond the scope of the book. Nevertheless, the lack of program listings for PETs makes the book much less useful for PET owners.

My second complaint involves the price —$9.95. Although the book is 77 pages long, much of this is taken up by wide margins and large diagrams.

But when all is said and done, *Introduction to Low Resolution Graphics* is an excellent work. I recommend it to anyone who wants to fully use the graphics abilities of his microcomputer.

Gary Sabot
Roslyn, NY

The Programmer

A Novel by Bruce Jackson
Doubleday & Company, Inc.
Garden City, NY
Hardcover, 281 pp., $8.95

Eddie Argo works as a computer programmer for the city of Buffalo. Eddie's ho-hum job is only matched by his ho-hum life with a ho-hum wife.

Then his credit rating gets marred by a computer error at a large department store. Eddie decides it's time to fight oversight with program might.

He begins with a computer fix of his wife's parking ticket. He takes on a new identity: Craig Hemsworth, the Computer Bandit. No system anywhere can match this electronic Robin Hood. Eddie plays with the FBI, IRS and American Express files, striking blows for both Eddie and the masses. Even staunch politicians find their private funds fiddled with.

Life is now anything but boring for Eddie, in his van equipped with a portable computer terminal, a blue box and a CB radio. Meanwhile, FBI agent Francis Kelly becomes more perplexed at the growing number of UCIs (unexplained computer incidents). It couldn't be Eddie—the computer says he is dead.

The Programmer will provide you with some delightful hours of reading. But sorry—it's not a how-to book.

Douglas Jones
North East, PA

PERSPECTIVES

(continued from page 226)

just complaints. Maybe it's because once users put forth the tremendous mental effort required to get the hang of things, they feel complacent and satisfied. They no longer have the aggravation driving them into actually writing a letter to the editor. An article is still more difficult to hurdle.

Barry David Brown
President
The Datak Corporation
Sparks, NV

Satisfied Customers

From my own experience things are not quite as bad as you would have your readers believe ("Word Processor Woes," p. 7, November 1980). Aspen Computer Service, Inc., is a service bureau which began business in 1974 with a Burroughs L8400 minicomputer. In 1977, in addition to our service bureau work, we became a dealer for The Digital Group (which went out of business in 1979). We now use four 50K Digital Group computers with dual single-sided, single-density disk drives, and two 300 LPM model 40, Teletype printers.

We are currently billing about $6000–$9000 a month in services and are not selling any hardware. We employ two full-time computer operators, and one part-time. We have written our own software with which we are very satisfied.

Our initial programs were written in Administrative Systems, Inc., OPUS II, which is a block structured extended BASIC. The object programs are semi-compiled. We are now writing our programs in Pascal (Michael Lehmans: Pascal/MT), which is compiled. We are very impressed with the operating speed. As a basis of comparison, the sorting of 1000 128-character records under OPUS II takes about 75 minutes; under Pascal about two minutes. The printing of a payroll check under OPUS II (with all the reading, calculating, updating and writing) takes about 20 seconds; under Pascal about five seconds. We are in the process of rewriting all our systems in Pascal.

Comparing our Burroughs L8400 to our DG systems, we have found that the latter outperforms Burroughs in programming ease, printer speed, price and maintenance cost.

So here you have a business using microcomputers successfully and happily!

Don A. Voltmer
President
Aspen Computer Service, Inc.
Aspen, CO

✓Reader Service—see page 210

Some Like It Not Some Like It a Lot

Three Views On Word Processing

Paradoxical Processing

Your comments about word processing in the November issue ("Word Processor Woes," p. 7) touched one of my nerves, resulting in these comments. Several months ago my office took delivery of an NBI 3000 Word Processing System. This electronic phenomenon has produced more paradoxes than I possibly could have foreseen. For example:

The system produces documents of unbelievable quality, mostly due, however, to the excellent Diablo printer.

The turn-around time for a normal two-page proposal letter, from dictation to final document, has not decreased, but rather has almost doubled.

Both my secretary and I appreciate the freedom associated with the easy editing and correction capabilities, which has resulted in more relaxed and productive dictation and transcription sessions.

The operator's guide is well written, and my secretary was able to become quite proficient on the system in a short time. But the guide serves more as a training guide, rather than a reference manual. Consequently, finding how to do something that isn't done very often is difficult and time consuming.

Every letter now goes through a "draft" stage, whereas many letters used to get typed only once, and then signed and mailed.

Many difficult typewriter functions, such as centering, lining up columns of numbers, aligning decimal points, printing equations and producing outlines, have been simplified to a few keystrokes on the word processor. This, however, is offset by the fact that many easy typewriter functions, such as the "hanging indents," have become more complicated or time consuming.

My secretary likes the idea of using a word processor, but has found that it doesn't replace a typewriter, which is still needed for many pre-printed forms or short notes.

The confounded machine picks the absolutely worst times to fail. It always happens at 3 PM as the almost-final draft of a 12-page document (that must be mailed today) is being printed.

The machine, as with most word processors on the market today, is too large to fit on the typewriter table of my secretary's desk. We had to get a special typewriter desk extension built.

We have used the word processing system to put together a rather professional-looking newsletter, as well as to do some mass mailings with personalized letters. In both of these applications, the system has greatly simplified the task.

While the list could go on, I think it sufficiently expresses my frustrations. I am pleased with the professional quality that word processing lends to my correspondence and documents. I like the flexibility that word processing offers. I think the per-station costs are too high, but these will come down sharply over the next couple of years. I have a difficult time accepting the fact that while some tasks became easier, many other, more often used tasks became more cumbersome. But, no matter how well the letter is printed, regardless of the technology used to produce the document, in spite of the time and effort invested, the Post Office will still manage to lose it, mis-deliver it or destroy it!

Roland K. Smith
Datapoint Eastern Region
Sales Manager
Cleveland, OH

Satisfaction Supreme

In light of your less than satisfactory experiences with word processing ("Word Processor Woes," p. 7, November 1980), I thought I'd answer your feedback request and give you the benefit of hearing from one who is supremely satisfied with the system he bought.

I purchased an Alpha Micro AM-1031 computer system with a Qume KSR printer and a Micro-term MIME-2A CRT. I forked over about 21 kilobucks, which is a bit more than the average hobbyist can handle. The main lump in my cost is the Control Data 9427H "Hawk" hard disk cartridge drive. A serious hobbyist could buy a moderately priced AM-100 CPU system with all of the available AM languages and utility programs, but hooked up to a floppy or Winchester storage unit, an Anadex or Paper Tiger and a Soroc IQ-120. This system should come in under $10K.

It took me about two intensive and sometimes heartbreaking days to learn the rudiments of VUE, Alpha Micro's text editor program, and I was then on my way. I've come a long way since then and am now completely at home with the system.

The only thing that I found painful about the AM system was that their manuals all assumed that you knew quite a bit about computers and the operating characteristics of large (IBM, DEC) computing systems. The AM is, I later found out, set up just like those big-boy mainframe systems. If you have had a thorough grounding in big system operation, you'll have no problem adjusting to the Alpha Micro system. The only equipment failure I had was a 7805 regulator and two switching regulator transistors in my QUME printer.

You've mentioned that you get very few letters or articles from satisfied users,

(continued on page 225)